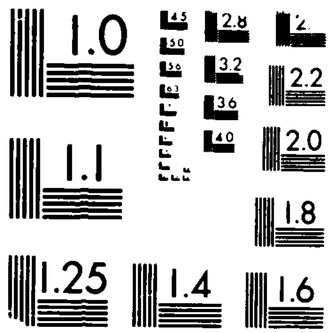
AD-A190 574 EFFECTIVELY CONTROLLING DATAGRAM CONGESTION ON THE DOD INTERNET SYSTEM GATEWAYS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI B J SCHOFIELD 1/2

UNCLASSIFIED DEC 87 AFIT/GE/ENG/87D-57 F/G 25/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

EFFECTIVELY CONTROLLING DATAGRAM CONGESTION ON THE DOD INTERNET SYSTEM GATEWAYS

THESIS

Bruce J. Schofield
Captain, USAF

AFIT/GE/ENG/87D-57

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/87D-57

EFFECTIVELY CONTROLLING DATAGRAM CONGESTION

ON THE DOD INTERNET SYSTEM GATEWAYS

THESIS

Bruce J. Schofield
Captain, USAF

AFIT/GE/ENG/87D-57

Approved for public release; distribution unlimited

AFIT/GE/ENG/87D-57

EFFECTIVELY CONTROLLING DATAGRAM CONGESTION ON THE DOD INTERNET SYSTEM GATEWAYS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Bruce J. Schofield, B.S.

Captain, USAF

December 1987

Approved for public release; distribution unlimited

Preface

This study was suggested and sponsored by Mr James Tontonoz of the Defense Communications Engineering Center (DECE). I appreciate his initial efforts to get me started and his continueing efforts to locate traffic data on my behalf. In addition, I appreciate the assistance Mr Ed Cain, also of DCEC, provided me when I asked him for help.

My thanks also go to my patient advisor, Lt Col Garcia, and my entire thesis committee, Major John Stibravy, Captain Wade Shaw, and Dr P. Nagarsenker for their support.

Most importantly, I want to thank my wife and children for staying with me during this thesis effort -- I couldn't have done it without you.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Abstract

The DoD Internet system consists of more than 20 constituent networks interconnected through the use of standard gateways and a standard set of Internet protocols. Constituent networks generally differ in transmission media and they may also be incompatible in terms of packet size, address format, speed, delay, and reliability.

Under the current implementation of the DoD Internet, a gateway's response to congestion is to discard datagrams. Discarding datagrams increases message delay and wastes network resources. Several congestion control methods have been proposed to improve the performance of the Internet. This study looked at two; Nagle's Fair queueing and Zhang's Metered queueing.

Nagle proposes to replace the single queue per outgoing channel with multiple queues, one for each source with datagrams passing through the gateway. Datagrams are removed from these queues one at a time in a round robin fashion. This procedure ensures each source is allotted a fair share of the channel bandwidth. The study found, through simulation, that this method insulated well behaved host from the presence of a badly behaved host. Badly behaved host are in effect punished through increased delay while well behaved host receive their fair share of the network resources. This researcher recommends Nagle's method be implemented for testing on the Internet.

Zhang proposal is basically a feedback method of congestion control. This method allows a gateway to control the rate at which host send datagrams through the gateway. This requires modification to the IP modules in the hosts and gateways and modification to the Source Quench message. These modifications will allow the gateways to sense traffic levels and to tell the host what rate to transmit at and for how long. However, Zhang did not define two parameters which are critical to the performance of her method. Both of these parameters depend on the Internet traffic profile which is not known at the present. Because these parameters are not defined, this study could not simulate the performance of Zhang's method. However, this researcher does recommend Zhang's method for future study.

EFFECTIVELY CONTROLLING DATAGRAM CONGESTION ON THE DOD INTERNET SYSTEM GATEWAYS

I. Introduction

Background

During the late 1960's, the Department of Defense, through the Defense Advanced Research Projects Agency (DARPA), sponsored the development of an experimental, packet switched computer network. This network, the Advanced Research Projects Agency Network (ARPANET), first became operational in 1969. By 1975, the ARPANET had developed to the point it had become an operational network. In 1975, control of the ARPANET was transferred from DARPA to the Defense Communications Agency (DCA).

The ARPANET was the first major network to be developed using packet switched technology (4:307). With the success of the ARPANET, a number of other networks were soon developed in both the military and private sectors. Some of these packet networks are terrestrial based systems like the ARPANET while others involve a variety of transmission media, such as satellite, local area networks, and mobile packet radio (4:307). Each of these systems was developed to meet a specific requirement; therefore, besides differing in transmission media, the networks may also be incompatible in terms of packet size, address

format, speed, delay, and reliability (27:113). However, as different as these networks may be, they must interoperate, especially the military networks (4:309). The DARPA research community recognized the need for diverse packet switched networks to interoperate and as a result, the DARPA Internet system has evolved over the last 10 to 12 years.

The DARPA Internet system is one of the original interconnected groups of networks (27:111). The Internet consists of more than 20 constituent networks interconnected in a general distributed fashion through the use of standard gateways and a standard set of Internet protocols (4:309;27:113). Figure 1 illustrates this concept.

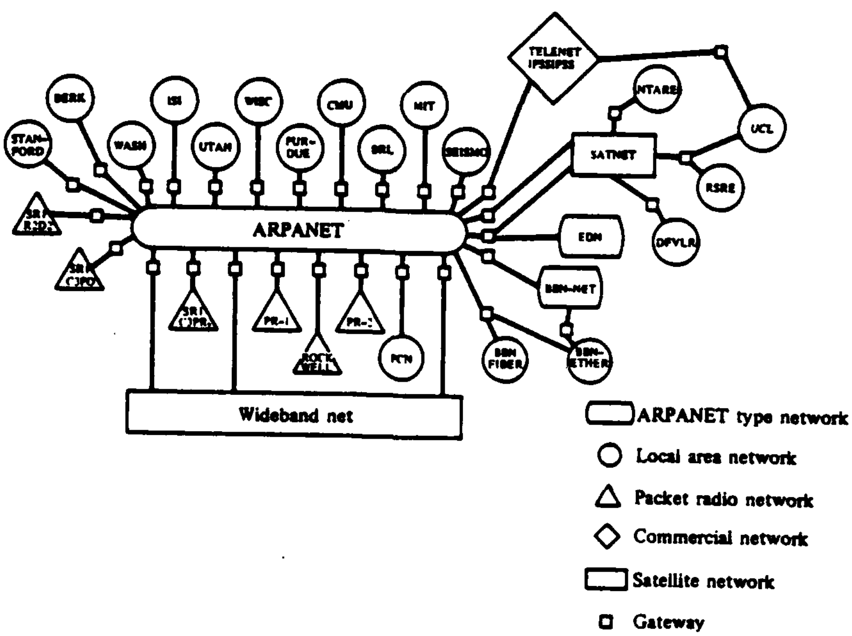


Figure 1. Internet Concept (29:450)

Under the current DoD Internet architecture, a gateway's response to overload conditions is to discard packets and send a source quench message to the host that was the source of the discarded packets. Hosts receiving the source quench messages are expected to use some reasonable scheme to reduce the traffic they send. However, there are several reasons the source quench mechanism is ineffective for congestion control. For example, the host receiving the source quench message may not be the root cause of the congestion problem. Furthermore, the appropriate response of a host receiving the source quench message has never been fully defined nor standardized.

Researchers from the Internet Research community have recently proposed two new methods for controlling congestion in the Internet. Nagle's Fair Queueing is the first of these methods. The objective of Nagle's Fair Queueing algorithm is to ensure that, despite the presence of badly-behaved hosts, well-behaved hosts receive their fair share of channel bandwidth. That is, at a minimum, a host should receive a share of the channel bandwidth which is inversely proportional to the number of hosts using the switch at that particular time (12:7).

Zhang's Metered Queueing method is the second proposed method. This method is based on the assumption that a feedback congestion control system is feasible in the Internet environment (31:3). Zhang proposes to modify the existing source quench message so that it provides specific control information to the host that receives it (31:4).

Problem and Objectives

The Defense Communications Engineering Center, the principle engineering activity of the Defense Communication Agency and sponsor of this thesis, is faced with the problem of determining how to effectively control the datagram congestion in the Internet gateways. John Nagle and Lixia Zhang, two researchers from the Internet research community, have proposed algorithms for controlling this congestion. The objective of this thesis is to determine whether either of these two methods can effectively control datagram congestion in the gateways. This determination will be made by simulating the performance of each of the two methods using a computer software model of the gateway's operational characteristics and the Internet's traffic profile.

Scope

This study is concerned only with congestion control in the Internet gateways. Specifically, this study focuses on the current and two proposed methods of controlling congestion on the Internet gateways.

General Approach

This thesis begins with a study of the architecture and protocol of the Internet system. Then, gateway traffic data is analyzed to

determine the characteristics of the traffic profile and the congestion problem. From this data, a model of the traffic profile is developed. Next, Nagle's Fair Queueing and Zhang's Metered Queueing algorithms are studied, modeled, and analyzed through simulation. Then, each of the two proposed algorithms is evaluated using the traffic profile model developed from the traffic data. This evaluation is accomplished through simulation. Finally, this thesis documents the models and techniques used during the evaluations and makes recommendations on the use of the proposed algorithms in the DoD Internet system.

Sequence of Presentation

The Internet system is composed of a variety of networks interconnected by gateways. Chapter 2 begins with a brief study of these networks and the gateways that interconnect them. Next, the various protocols which govern the operation of the Internet are examined. Finally, Chapter 2 presents a brief study of the Internet traffic.

Chapter 3 looks at methods of controlling congestion and begins with the Internet's Source Quench method. Next, Nagle's Fair Queueing algorithm is examined. The chapter concludes with an analysis of Zhang's Metered Queueing.

Chapter 4 discusses the development of the models used in the simulations. This chapter begins with a discussion the assumptions upon which the models are based. Then, Chapter 4 presents the Traffic model and the model for the Internet system.

The results of the simulations conducted using these models are analyzed in Chapter 5. Conclusions and recommendations are presented in Chapter 6.

II. The Internet System

Introduction

This chapter presents the Internet system. The characteristics of the different classes of networks which make up the Internet are described. Then, the functions and operation of the gateways which connect these networks are discussed. The protocols that govern the operation of the Internet system are examined next. Finally, the characteristics of the Internet traffic are presented.

For the benefit of the reader who is not familiar with the concept of internetworking, this chapter begins with a brief discussion of the approaches to internetworking.

Internetworking

The purpose of internetworking is to allow hosts, connected to different networks, to communicate. There are two different approaches to interconnecting networks. One approach is connection-oriented and involves the interconnection of virtual circuits, while the other provides connectionless (datagram service) between the networks.

X.75. The International Telegraph and Telephone Consultative Committee (CCITT) developed X.75 as its specification for the interconnection of public data networks using its X.25 protocol. The

CCITT's X.25 protocol provides virtual circuit service and is "perhaps the best known and most widely used protocol standard" for packet switched networks (29:420).

The X.75 interconnection takes place at the node level. Thus, in addition to the packet switching nodes of a network, each network which is to be interconnected has an additional device referred to as a Signalling Terminal (STE)(Figure 2). The interface between STE's is specified by X.75 and is very similar to X.25 specification for the interface between a host and a packet switching node (16:516).

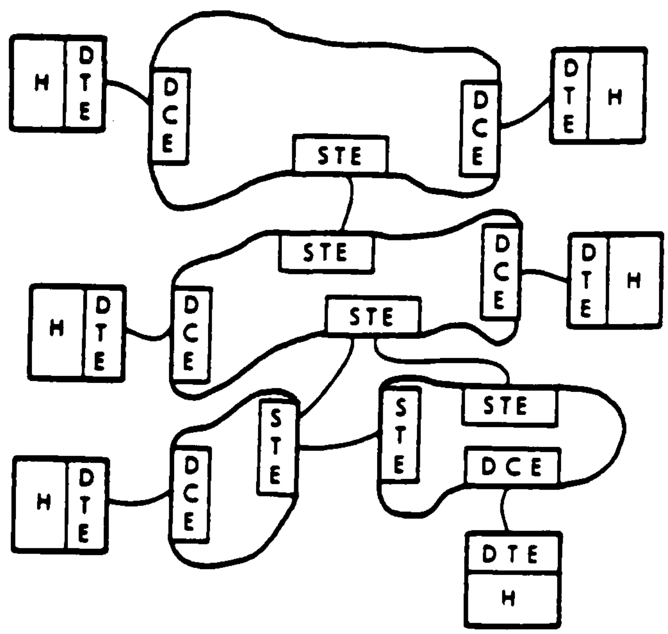


Figure 2. X.75 Interconnection

The end result is a series of virtual circuits which span the networks separating the two hosts (Figure 3). Each individual virtual circuit is bounded and controlled by the network it spans (16:517; 29:441). However, when these individual connections are linked together by X.75, they appear to the two hosts as a single virtual circuit between them (29:441).

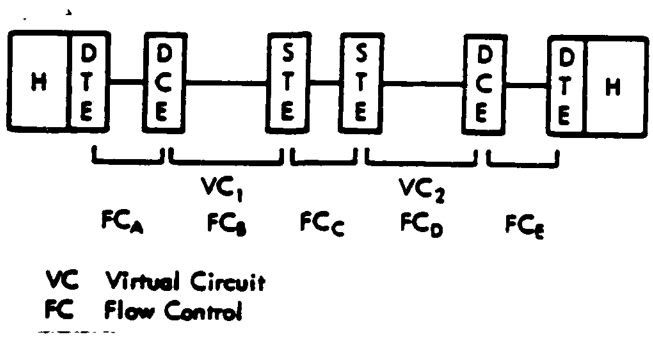


Figure 3. X.75 Transmission Path (16:518)

Internet Protocol. The alternative to CCITT's virtual circuit approach is to provide datagram service between the connected networks. This is the approach the DoD's Internet Protocol (IP) takes. The Defense Advanced Research Agency (DARPA) first developed IP in support of the Internet Project sponsored by the DoD in the mid 1970s (4:307). Since then, the DoD has standardized IP (29:441).

IP differs from X.75 in two important ways. First, since IP provides datagram service, it must rely on a common transport layer protocol to ensure reliable end-to-end service. A common transport layer protocol is not necessary with X.75 because it provides virtual circuit service between the connected hosts.

Second, IP interconnects networks at the host level using gateways, whereas X.75 interconnected networks at the node level. Gateways, under the Internet architecture, are devices which appear as hosts on two or more networks (Figure 4). These gateways make it possible for IP to interconnect networks with different access protocols, while X.75 required the networks to implement X.25 (27:113).

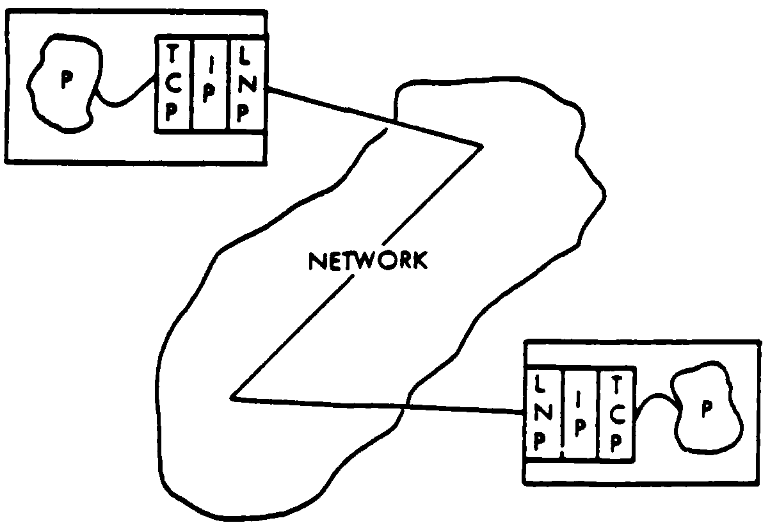


Figure 4. IP Interconnection (16:519)

These and additional differences in IP and X.75 are listed in Table I. Since IP architecture is fundamental to this research project, the remaining sections of this chapter deal with its constituent networks, gateways, and protocols in more detail.

Table I. Comparison of IP and X.75 (29:442)

IP	X.75
Host-level gateway	Node-level gateway (STE)
Datagram Service	Virtual Circuit Service
Gateway must know IP, two network access schemes.	Gateway must maintain state information about all virtual circuits.
Adaptive routing easily implemented.	Fixed routing typically; adaptive routing more difficult.
All host must have IP, may need common layer 4.	All networks must be X.25

Constituent Networks

The Internet system is a collection of heterogeneous networks interconnected in a manner which allows a host on one network to communicate with a host on another network. The networks which collectively form the Internet system will generally fall into one of two categories; wide-area or local-area networks. However, because of the proliferation of local-area networks, the Internet architects have introduced subnets as a third category.

Wide Area Networks. Wide-area or long-haul networks generally cover a large area and connect hosts which are widely dispersed. These networks may be very complex (e.g. the ARPANET) or simple point-to-point networks (3:3).

Local Area Networks. In contrast to wide-area networks, local-area networks cover a relatively small geographical area. For example, local area networks may be used to connect computers within a single building or on a college campus. In addition, the local area network's data transfer rates are generally higher and delays generally lower than those found in wide-area networks (3:3). There are numerous varieties of local area networks; however, most are based on the ring or bus topology.

Subnets. The concept of subnets allows an organization with a complex system of many interconnected local area networks (LANs) to maintain the identity of each network while protecting the Internet System "against explosive growth in network numbers and routing complexity" (3:5). The subnet extension essentially hides the complex LANs system from the rest of the internet.

Gateways

"The concept of a gateway is common to all network interconnection strategies" (5:1392). While the primary purpose of a gateway is to interconnect two or more networks, a gateway may also perform routing or protocol translation (1:27). Figure 5 illustrates the general structure of a gateway.

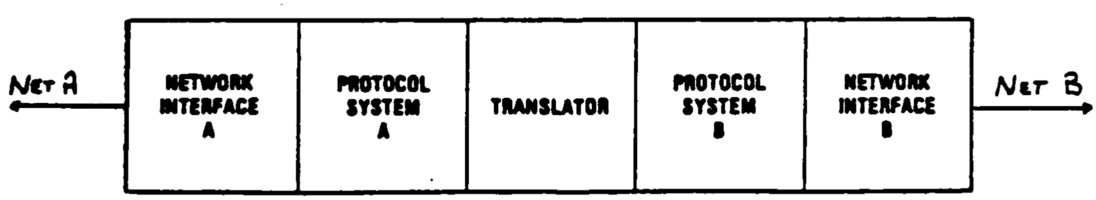


Figure 5. Gateway Structure (1:27).

This structure provides an interface to each of the networks the gateway is connected to. The structure also includes a protocol module for each of the networks. These protocol modules are connected by a module which is capable of translating either protocol into the other. A gateway based on this structure is capable of connecting similar or dissimilar networks. Postel describes two different type of gateways conforming to this general structure (16:513-515).

Protocol-translation Gateway. The first type of gateway Postel describes is the "protocol-translation" gateway. This type of gateway translates between the different protocols used by the networks it interconnects. For example, if the gateway receives a message from a host on network A which is addressed to a host on network B, the gateway replaces the message with a different message having the same meaning but satisfying the protocol syntax of network B (16:514).

Media-conversion" Gateway. The "media-conversion" gateway is the second type of gateway Postel describes. This type of gateway is based on the concept of encapsulation. This means the message unit (header and data) of a higher level protocol is treated as data by the lower level protocols. For example, a layer 3 protocol can encapsulate the message unit of a layer 4 protocol by attaching its layer 3 header and trailer to the layer 4 message as shown in Figure 6.

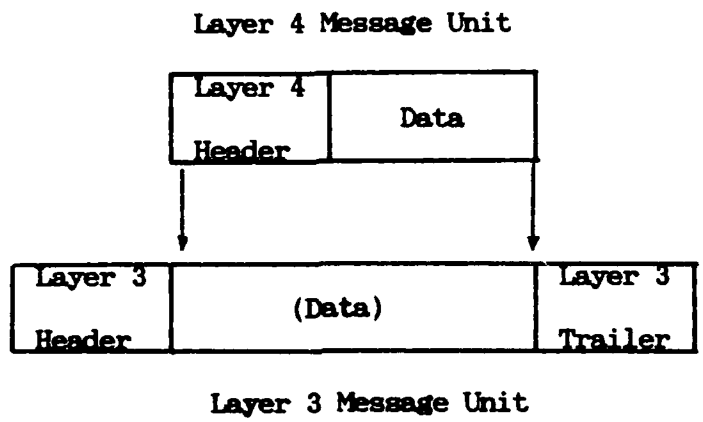


Figure 6. Encapsulation

As a media-converson gateway receives packets from network A, it strips off the header and trailer network A attached to the message. Then, the gateway reads the header of the message to determine the message's destination. It uses this information to determine the destination on network B. Next, the gateway builds a packet packet header using this routing information and attaches it to the message. Finally, the gateway passes this packet to the network interface module to send over network B.

In comparison, the protocol-translation gateway is more complex than the media conversion gateway. The protocol-translation gateway relies on common lower level (Layers 1 and 2) protocols in order to translate between different upper level (Layer 3 and above) protocols. On the other hand, media-conversion gateways rely on a common upper

level protocol to convert between different lower level (Layers 1 and 2) protocols (16:514). This allows the media-conversion gateways to connect networks which use different transmission media (16:514). For example, a media-conversion gateway may be used to interconnect two networks; one which uses land lines as a transmission media while the other uses packet radio.

<u>The Internet Gateway</u>. The DoD Internet system uses standard gateways of the media-conversion type to interconnect a collection of heterogeneous networks. Each gateway is connected to two or more networks as if it were a host on each (14:1-2). The main purpose of these gateways is to receive internet datagrams from one network and forward them on another toward their final destination. To accomplish this task, each gateway (and all hosts) implements a common protocol (Internet Protocol) and assumes each adjacent network is using the same host-to-host protocol (14:1-2). In addition, each Internet gateway must perform several basic functions; such as, interfacing to local networks, routing, fragmentation, and error reporting. The following paragraphs discuss these functions.

<u>Interfacing</u>. As a media conversion type of gateway, an Internet gateway makes use of the Internet Protocol which is common to all gateways and all hosts connected to the networks which comprise the Internet system. In order to interface two networks, an Internet gateway must make use of the concept of encapsulation as explained

above. In addition, for each network it is connected to, a gateway must be capable of receiving, processing, and sending IP datagrams "up to the maximum size supported by that network, this size is the network's Maximum Transmission Unit or MTU" (3:7). Finally, a gateway must be capable of mapping the IP-datagram's destination address into an appropriate address for each network it is connected to (3:7).

Routing. The Internet system provides a global address which uniquely identifies each host connected to the Internet. The structure of the global address is hierarchical as Figure 7 shows (6:113-114).

{ Network Address, Local Address }

Figure 7. Internet Address (3:5)

Using this address, Internet gateways must be able to route each Internet datagram to its next destination. If the Network portion of the global address maps to one of the networks the gateway is directly connected to, then the gateway routes the datagram to the host identified by the local address. Otherwise, the gateway must route the datagram to another gateway. The gateways maintain routing tables for this purpose.

Fragmentation. Fragmentation is the process of dividing large datagrams into two or more smaller datagrams. This procedure is essential to the operation of the Internet system because the maximum transmittable unit (MTU) of some networks is smaller than that of others. A network's MTU is determined by its network access protocol. For example, networks using the ARPA network access protocol, BBN 1822, can accept messages of up to 8063 bits. However, it is possible that a network using the Ethernet access protocol can only accept messages of 256 bits. Therefore, before a gateway can route a message it receives from an ARPA network over the Ethernet, it must fragment the message into datagrams no larger than 256 bits. How the gateway fragments a datagram is governed by the Internet Protocol and is discussed in that section.

Error Reporting. Gateways must be able to recognize and respond to certain error conditions. These error conditions include congestion within the gateway, problems with the parameters in the datagram header, or destinations that are unreachable for some reason. How the gateway responds to these errors is a function of the Internet Control Message Protocol and is discussed in that section.

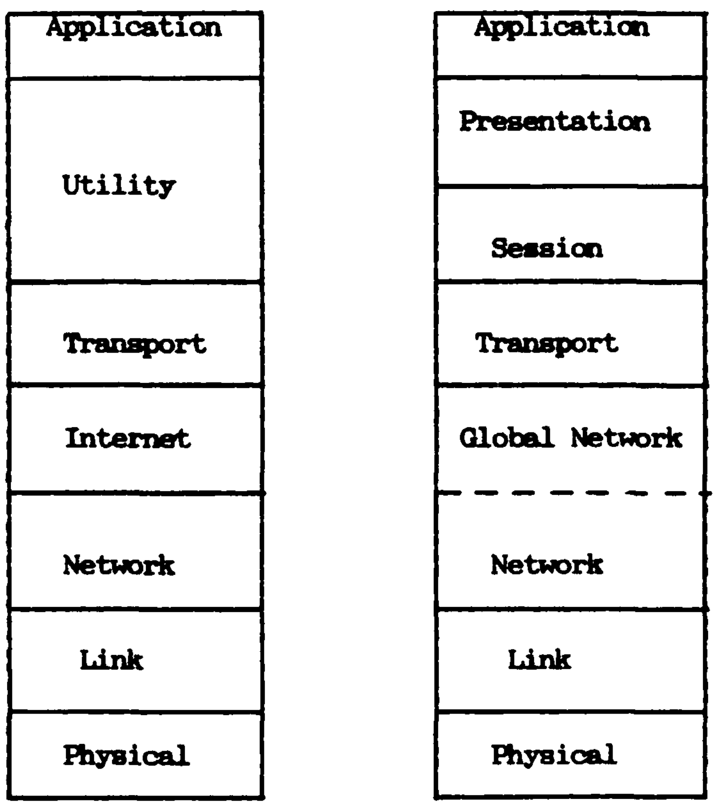
Protocols

The Reference Model of Open Systems Interconnection (OSI) developed by the International Standards Organization (ISO) is perhaps the most

widely publicized and accepted protocol architecture (4:309; 29:371). The OSI model is based on the structuring concept of layering (29:386). Padlipsky defines layering as:

> The control information of a given protocol must be treated strictly as data by the next lower protocol (with processes at the top and the transmission medium at the bottom) (14:16).

A second family of protocols grew out of the research conducted by the DARPA Research community on the ARPANET and internetworking. Like the OSI model, the DoD Architecture Model is also a layered model. Figure 8 shows how these two models compare.



DoD Internet Model ISO Model

Figure 8. DoD and ISO Protocol Architecture Models (4:310)

From Figure 8, several differences in the two models are apparent. At the higher levels, the ISO model provides distinct session and presentation layers while the DoD model lumps these functions into a single utility layer. On the other hand, the DoD model provides a distinct internet layer while the ISO model splits the network layer into two sublayers; with the global network sublayer responsible for internetworking. This fact may be a consequence of the differing approaches to internetworking. The DoD Internet Model is designed to interconnect heterogeneous networks; whereas, the ISO model assumes more homgeneity (4:309). By providing a separate internet layer, the DoD model places additional emphasis on internetworking and isolates the transport and network layers from the problems associated with internetworking.

Figure 9 identifies the relationships between the various protocols which comprise the DoD Internet Protocol Hierarchy. The following paragraphs briefly explain each of the layers as well as the more important protocols.

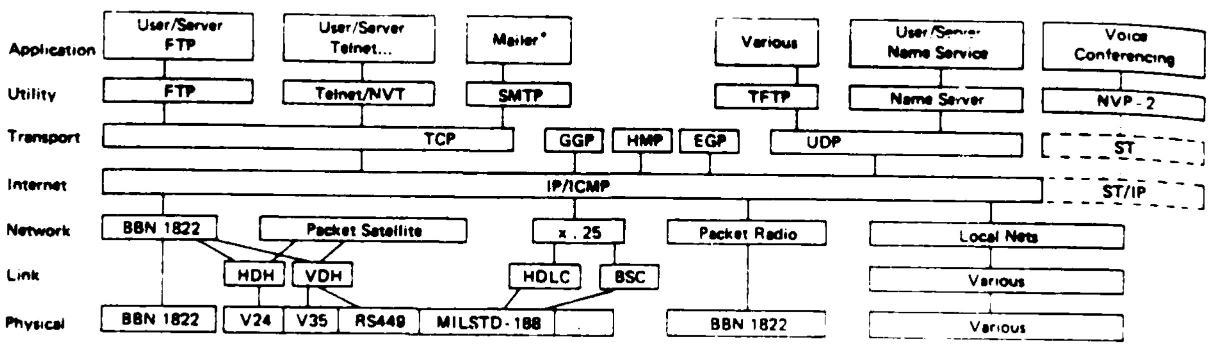


Figure 9. DoD Internet Protocol Hierarchy (4:312)

Application Layer. This, the highest level, collectively represents the processes which are responsible for initiating and terminating all communications. Application layer processes rely on the utility layer to provide the functions necessary to transfer data.

Utility Layer. The protocols at this layer are designed for specific applications such as resource sharing or remote access. For example, the File Transfer Protocol (FTP) (22) is intended to transfer files between two processes. TELNET (21) is another example of a utility layer protocol. TELNET allows all remote terminals to connect to hosts as standard "Network Virtual Terminals" (4:313). Other Utility layer protocols are Simple Mail Transfer Protocol (20), Trivial File Transfer Protocol (28), Name Server Protocol, and Network Voice Protocol (4:313).

Transport Layer. The primary purpose of the transport, or host-to-host, layer protocols is to transfer data between processes on two different hosts. These protocols may or may not provide reliable service. In fact, Stallings sees the need for four different types of protocols at this level (29:399).

1. A connection-oriented protocol is need to provide reliable, sequenced exchange of information.
2. A connectionless, or datagram, protocol is needed to provide low overhead service to those higher level protocol which ensure their own reliable service.

3. A speech protocol is need to transfer a stream of data with minimum delay.
4. A data protocol that combines the capabilities of a connection-oriented protocol and a speech protocol is required to satisfy the requirements of real-time communication.

The DoD Internet Model includes three primary protocols at the transport layer; Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and ST protocol. The TCP is a connection-oriented protocol which provides reliable end-to-end service. The Deputy Undersecretary of Defense for Research and Development has declared TCP "to be a basis for DoD-wide inter-process communication protocol standardization" (19:1). Because of its importance to the Internet and this research project, a clear understanding of TCP is essential. Therefore, the author is including, in this chapter, a section which discusses TCP in detail.

The UDP provides datagram service to those applications which do not require the quality of service TCP provides. The ST protocol is an experimental protocol being designed to support the broadcast, multicast, and conferencing services that do not require highly reliable service but do require minimum delay (4:313).

In addition to these three host support protocols, the transport layer of the Internet includes several protocols that either support the operation of the gateways or monitor the operation of the hosts. The Gateway-Gateway Protocol (GGP) and the External Gateway Protocol are protocols which support the exchange of gateway routing and status

information. GGP is specified by RFC 823 and the External Gateway Protocol is specified by RFC 904. The Host Monitoring Protocol (HMP) allows the monitoring of the hosts connected to the Internet System and is specified by RFC 869 (7).

<u>Internet Layer</u>. The DoD Internet architecture is based on a standard protocol at the internet layer. This protocol, the Internet Protocol (IP), provides internet addressing, routing, and error control. Because of the important role IP plays in the Internet and this research project, a clear understanding of this protocol is essential. To provide the reader with this background, the author has included, in this chapter, a section on the IP.

<u>Network, Physical, and Link Layers</u>. Protocols at these three layers define the interface between the host and the communications subnet. The DoD Internet Model does not specify these protocols. The DoD Internet systems accepts constituent networks as they are; therefore, these three layers of the DoD Internet Model merely recognize that these protocol layers exist. Some of the more common protocols at these three layers are identified in Figure 9.

<u>Transmission Control Protocol</u>. "TCP is a connection oriented, end-to-end reliable" transport layer protocol (19:1). The primary purpose of TCP is to provide highly reliable, securable communications between pairs of processes running on different hosts. The hosts may be

connected to the same network or to separate networks which are part of the Internet system. TCP assumes that each network which interconnects the hosts provides no more than "simple, potentially unreliable datagram service" (19:1). Because of this assumption, TCP must provide six services in order to provide its users with reliable, securable connection service; these services are discussed in the following paragraphs. This section concludes with a discussion of the interface between TCP and its upper and lower layers.

Services

Basic Data Transfer. TCP transfers data as "continuous streams of octets in each direction" (19:4,12). TCP divides the data it receives from its users into blocks. The maximum size of each block is specified by the TCP module at the destination host. From these blocks of data, the TCP forms a TCP-segment by prefixing the data with a TCP header. This header is 24 octets in length and contains information which is useful only to the TCP. In addition to the TCP header, each segment is conceptually prefixed with a pseudo header (19:16). These 12 octets are actually carried by the header of the Internet layer protocol. The pseudo header contains the source and destination addresses of the segment, along with other information. TCP also provides its users with a push function. The push function allows the source process to ensure all of its data, up to the time of the push, is promptly transmitted and delivered to the destination process (19:4,12).

Reliability. Padlipsky describes the degree of reliability sought during the development of TCP:

> Irrespective of the properties of the communications subnetworks involved in internetting, TCP is to furnish its users -- whether they be processes interpreting formal protocols or simply processes communicating in an ad hoc fashion -- with the ability to communicate as if their respective containing hosts were attached to the best communications subnet possible (e.g. a hardwired connection) (14:15).

This statement implies TCP must be able to detect and recover from lost or duplicate segments, segments arriving out of order, and transmission errors.

TCP uses a system of sequence numbers and positive acknowledgements to detect and recover from lost or duplicate segments and segments arriving out of order. The TCP-segment header includes fields for both a sequence number and acknowledgement number. Each of these fields is 32 bits long. These rather large fields are necessary because TCP sequences and acknowledges each octet of data instead of each segment (19:24).

Sequence numbers are not tied to a global clock; therefore, each TCP includes an initial sequence number generator. The purpose of this procedure is to ensure that for any particular connection, sequence numbers due to the present connection do not duplicate sequence numbers that may still exist from a previous connection.

Since TCP connections are full-duplex, each connection requires a send sequence number and a receive sequence number. The two TCPs must synchronize these sequence numbers (i.e., establish the connection) before they can use the connection to exchange data. A connection is

established through the three-way handshake process illustrated in Figure 10.

```
TCP A ------> TCP B   SYN   My sequence number is X
TCP A <------ TCP B   ACK   Your sequence number is X
                      SYN   My sequence number is Y
TCP A ------> TCP B   ACK   Your sequence number is Y
```

Figure 10. Three-way Handshake (19:27).

Special segments known as TCP control segments are used during this process. Control segments carry a control bit in their header which identifies them as SYN, ACK, or SYN-ACK segments. TCPs only use the SYN and the SYN-ACK segments to establish a connection; however, the ACK also serves an important function during the transfer of data over the connection. Since the source TCP assigns a sequence number to every octet of data, each octet of data must be acknowledged. However, the destination TCP will not acknowledge the data unless it is sure no errors occurred during transmission.

To detect transmission errors, TCP uses a checksum. The checksum is also carried in the segment header and covers the header, data, and

the pseudo header. Before it acknowledges a segment, the receiving TCP computes the checksum and compares it to the value of the checksum field in the segment header. If the two values are identical, then checksum is unable to detect any errors and the destination TCP can acknowledge the segment.

The destination TCP acknowledges the segment it has received by sending an ACK segment. It forms this segment by setting the ACK control bit of the header and placing the sequence number it expects to receive next in the acknowledgement number field of the header. Then, the TCP either attaches this header to data it has to send over the connection or, if necessary, sends the ACK segment without any data. Notice that the acknowledgement merely indicates to the source TCP that the destination TCP has assumed responsibility for the data; it does not imply that the data has been delivered to the destination process.

If the destination TCP discovers a transmission error, then it discards the segment without acknowledging it. Since it has not received an acknowledgement for the discarded segment, the sending TCP will retransmit the segment when its retransmit timer expires. Therefore, "as long as the TCPs continue to function properly and the internet does not become completely partitioned, no transmission errors will affect the correct delivery of data" (19:4).

Flow Control. TCP uses a dynamic window, controlled by the receiving TCP, to control the flow of data over a connection. With each ACK it sends, the receiving TCP includes the number of octets it is

prepared to receive in the window field of the header. This gives the sending TCP permission to send octets of data with sequence numbers within the window. As shown in Figure 11, the window begins with the acknowledge number and ends at the acknowledge number plus the value of the window field in the header.

```
         |                                           |
         |<------------- Window ------------->|
         |                                           |
    acknowledge                                 acknowledge
      number                                  number + window
```

Figure 11. TCP Window

However, before the sending TCP sends another segment it determines whether there are any unacknowledged sequence numbers which fall into the window. If so, it must reduce the number of octets of data it can send by this amount. Repeated applications of this procedure result in small windows. Therefore, RFC 793 suggests TCP implementations contain a procedure to combine small window allocations into larger ones (19:44).

Because the window is carried with the ACK, sending TCPs, with data to send, must periodically send a segment even though the window size is zero. This forces the receiving TCP to acknowledge the segment and report its current window. From this information, the sending TCP may learn that the window is still zero or that it has opened up.

Multiplexing. Several processes on a single host can require the services of the TCP at the same time. To accommodate multiple users, TCP employs a multiplexing scheme. The TCP assigns a port to each process using it. A socket is formed by concatenating the port address of a process with the internet addresses of the TCP. Since each process is assigned an individual port within a TCP and each TCP is assigned a unique internet address, a socket uniquely identifies a process throughout the Internet system. Therefore, a connection is explicitly defined by a pair of sockets.

Connections. TCP is a protocol which provides virtual circuit service at the transport layer. In general, communications with virtual circuit service can be divided into three distinct phases: Setup; Data Transfer; and Shutdown (30:188). A TCP connection passes through a sequence of states which span these three phases. Table II lists and defines the TCP connection states. The first three states correspond to the setup phase while the fourth state, established, represents the data transfer phase. The three-way handshake procedure (Figure 10) brings a TCP connection through the first three states to the established state. The remaining seven states equate to the shutdown phase.

For each of its connections, the TCP maintains a transmission control block (TCB). The TCB is a data structure containing all information pertinent to the connection. For instance, local and

Table II. TCP Connection States (19:21-22)

STATE	MEANING
Listen	Waiting for a connection request from any remote TCP and port.
SYN-Sent	Waiting for a matching connection request after having sent a connection request.
SYN-Received	Waiting for a connection request acknowledgement after having both received and sent a connection request.
Established	An open connection, data received can be delivered to the user. The normal state for the data phase of the connection.
FIN-Wait-1	Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.
FIN-Wait-2	Waiting for a connection termination request from the remote TCP.
Close-Wait	Waiting for a connection termination request from the local user.
Closing	Waiting for a connection termination request acknowledgment from the remote TCP.
Last-ACK	Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).
Time-Wait	Waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
Closed	No connection state at all.

remote socket numbers, security and precedence information, pointers to the user's send and receive buffers, pointers to the retransmit queues, the pointer to the current segment, connection state information, and several variables pertaining to the send and receive sequence numbers are all stored in the TCB (19:19).

A TCP connection changes states in response to events. TCP events fall into three categories: User Calls; Incoming Segments; and Timeouts. The significant user calls are OPEN, SEND, RECEIVE, CLOSE, ABORT, and STATUS. The important segments include those involved in the three-way handshake (i.e., SYN, ACK, and SYN-ACK) and those with the RST or FIN control flag (19:22).

Precedence and Security. In a military environment, it is imperative that the communications system prevent the compromise of classified data. In a communications network such as the Internet system, classified data could be compromised if the system delivered it to a user who was not authorized to receive data of that classification. To prevent such an incident, TCP allows its users to specify the security and precedence level of their communications. Once the security and precedence level are specified for a port, TCP will only allow this port to connect with a port of the same security level. Once a connection is established, communications will take place at the higher of the two requested precedence levels.

Retransmission Timeout. TCP relies on a timer to determine when to retransmit segments. The time interval of this timer must be determined dynamically to account for the wide variety of networks that form the Internet system (19:41). RFC 793 provides this algorithm:

> Measure the elapsed time between sending a data octet with a particular sequence number and receiving an acknowledgment that covers that sequence number (segments sent do not have to match segments received). This measured elapsed time is the Round Trip Time (RTT). Next compute a Smoothed Round Trip Time (SRTT) as:
>
> SRTT = (AlPHA * SRTT) + ((1- ALPHA) * RTT)
>
> and based on this, compute the Retransmission Timeout (RTO) as:
>
> RTO = min[UBOUND, max[LBOUND, (BETA * SRTT)]]
>
> where UBOUND is an upper bound on the timeout (e.g., 1 minute), LBOUND is a lower bound on the timeout (e.g., 1 second), ALPHA is a smoothing factor (e.g., 0.8 to 0.9), and BETA is a delay variance factor (e.g., 1.3 to 2.0) (19:41).

Interfaces. In the DoD Internet system, TCP interfaces with processes at the higher layer and with the Internet Protocol at the lower layer.

User to TCP Interface. TCP is assumed to be an operating system module. Therefore, users can access TCP through a set of calls (19:3,8,9). RFC 793 specifies six user calls. These calls allow the user to OPEN a connection, SEND and RECEIVE data, CLOSE or ABORT the connection, and to obtain the STATUS of the connection.

TCP to IP Interface. This interface is left unspecified in RFC 793. However, the interface is assumed to consist of two calls; one for sending data and another for receiving data.

Internet Protocol.

Functions. The Internet Protocol performs two basic functions which are essential to the performance of the Internet system; addressing and fragmentation.

Addressing. First, IP implements the addressing scheme which allows gateways to route the IP-datagrams toward their destination. As part of its header, each IP-datagram carries with it an internet address. This address is read at each gateway and used by the gateway to determine the next hop destination for the datagram. This procedure is very similar to the procedure used by packet switching nodes to determine the destination of a packet on a network.

Fragmentation. IP is required to deliver data to a destination Host-Host level protocol in the same form as the IP module at the source received the data (2:6). Therefore, if it is necessary to fragment a datagram as it traverses the Internet, then that datagram must be reassembled before IP can deliver it. The Internet architects considered two methods of reassembling fragmented datagrams. The first method requires the fragmented datagram to be reassembled as soon as

possible. This practice meant that any gateway that received a fragmented datagram would have to reassemble the datagram if the next-hop network could accept the datagram in one piece. This method presents two problems. First, it introduces the possibility of reassembly lock-up at gateways. This form of lock-up would result whenever all the buffers at a gateway are occupied by fragments waiting to be reassembled. However, since there are no free buffers, the gateway can not accept those fragments necessary to complete the reassembly of the datagrams. Second, IP provides datagram service which means that each datagram (or fragment) is independently routed through the Internet toward its destination. Therefore, all fragments of a datagram may not pass through the same gateway, thus making reassembly impossible.

In light of these problems, the Internet architects decided to reassemble fragmented datagrams only at their destination. This decision means that reassembly resources are only required at destination IP modules and not at gateway IP modules. These resources consists of a "data buffer, header buffer, fragment block bit table, total data length field, and a timer" (17:27). To reassemble a fragmented datagram, the IP looks for fragments which have common values in their identification, source, destination, and protocol fields. Then, it places the data portion of these fragments in the relative position indicated by the data offset field contained in the IP-header. The first fragment of a datagram is identified by an offset of zero while the last fragment will have a zero bit as its more fragment flag

(17:9). If the reassembly process is not completed by the time the timer runs out, the datagram is discarded.

Discarded datagrams severely affect the performance of the Internet system. Prue provides the following example to illustrate this point:

> Examine what happens when a window is 35 datagrams wide with an average round trip delay of 2500 msec usong 512 byte datagrams when a single datagram is lost at the beginning. Thirty five datagrams are given by TCP to IP and a timer is started on the first datagram. Since the datagram is missing, the receiving TCP will not send an acknowledgement, but will buffer all 34 of the out-of-sequence datagrams. After 1.5 X 2500 msec, or 3750 msec, have elapsed the datagram times out and is resent. It arrives and is acked, along with the other 34, 2500 msec later. Before the lost datgram we might have been sending at the average rate a 56 Kbps line could accept, about one every 75 msec. After loss of the datagram we send at the rate of one in 6250 mseo, over 83 times slower (25:9).

IP places some restrictions on the maximum and minimum size of the IP-datagrams. RFC 791 specifies that "every Internet destination must be able to receive a datagram of 576 octets, either in one piece or in fragments to be reassembled" (17:25). In most cases, this maximum is sufficient to allow the transfer of data in 512 octet blocks (23:266).

As a consequence of the IP fragmentation procedure, every IP-module must be able to forward an IP-datagram of at least 68 octets. This size is fixed by the fact that each IP-datagram consists of a header and data. The maximum size of an IP-header is 60 octets and the minimum size of a data fragment is 8 octets (17:25). This also means that any network connected to the Internet must also be able to accept and deliver a message of at least 68 octets. However, this requirement does not preclude a network from fragmenting and reassembling a datagram

within its boundaries as long as this procedure is transparent to IP and its upper level protocols. Such fragmentation is often referred to as "intranet" fragmentation.

Mechanisms. IP includes four mechanisms which are essential to the datagram service IP provides its user. Each of these is determined by the user and passed to IP along with the data as parameters of the user's call to IP. IP incorporates these parameters into the datagram's header so they are available to each IP module that processes the datagram.

Type of Service. Type of Service is the first of these mechanisms and allows the user to specify the quality of service desired. RFC 791 describes the Type of Service as "an abstract or generalized set of parameters which characterize the service choice provided in the networks that make up the Internet" (17:2). Some of the networks which make up the Internet may provide several different grades of service while others provide just one. Gateways use the Type of Service parameter provided by the user to determine the grade of service to request from those networks which provide options. These options typically allow the network user to request different levels of precedence, reliability, and delay and throughput (29:459). For example, the type of service parameter of a datagram may indicate to

the gateway that delay should be minimized for this datagram. On the other hand, the next datagram this gateway processes may require maximum throughput.

Time to Live. Time to Live is the next mechanism used by IP. The purpose of the Time to Live mechanism is to ensure undeliverable datagrams are discarded. IP accomplishes this by establishing an upper bound on the lifetime of a datagram. This bound is necessary because some higher level protocols make the assumption that if a datagram is to reach its destination it will do so within a certain period of time. Using the time to live mechanism, IP is able to provide the upper level protocols with this assurance.

When an upper level protocol issues a send call to IP, one of the parameters it passes along with the data is the value for the time to live parameter. IP places this number in the TTL field of the header it attaches to the data as it builds the IP-datagram. As this datagram traverses the Internet, each IP module that processes the header must decrement the TTL count by at least one. However, if a gateway should hold the datagram for more than one second then it must decrement the TTL count by the number of seconds it held the datagram. Although the TTL parameter is meant to represent the maximum lifetime of a datagram in seconds, it is often interpreted to represent the maximum number of network hops a datagram can make before it reaches its destination because a gateway normally does not hold a datagram for more than one second (3:36; 29:459). If the TTL count reaches zero before a datagram

is delivered to the upper level protocol at its destination, then the IP module processing the datagram at that time must discard it (17:2).

Options. The third IP mechanism is the Options parameter. The purpose of the options parameters is to provide the control functions necessary to meet certain special communication requirements. For example, the options parameter can be used to attach a security classification label to the datagram or to specify the route for the datagram

Header Checksum. The final mechanism is the Header Checksum. The purpose of the header checksum is to prevent an IP module from processing an IP header which contains errors. The header checksum is checked at each IP module before the header is processed. If the checksum is correct, the module continues to process the header. But, if the checksum indicates the header contains an error, the datagram is discarded. Since some fields in the header change (e.g. TTL field), the checksum must be recomputed after the header is processed. To reduce overhead, IP uses a relatively simple checksum which is easy to compute. However, although the checksum is simple, experiments have shown that it is adequate.

Internet Control Message Protocol. The purpose the Internet Control Message Protocol (ICMP) is to allow the hosts and gateways connected to the Internet to exchange information, in the form of ICMP

messages, pertaining to the processing, routing, and flow of IP-datagrams. RFC 792, the document that specifies this protocol, states:

> The purpose of these control messages is to provide feedback about problems in the communication environment, not to make IP reliable. There are still no guarantees that a datagram will be delivered or a control message will be returned (18:1).

Although ICMP relies on services of IP for the transfer of these messages, ICMP is considered an integral part of the IP and must be implemented in every IP module (18:1). In general, ICMP messages are either error messages or information messages. However, all ICMP messages are sent as IP-datagrams. The data portion of this datagram contains the ICMP message and varies according to the type of message.

Error Messages. As part of the data portion of each error message, ICMP includes the first 64 bits of data from the IP-datagram it is reporting on. Assuming TCP is the transport layer protocol being used, then these 64 bits will help the port addresses of the source and destination processes. From this information, the host can determine which of its processes and connections originated the datagram. RFC 792 identifies five different types of error messages.

Destination Unreachable Message. A gateway may send this message to tell the source host the gateway could not forward the host's datagram. In addition, this message will also indicate whether the destination network or host was unreachable.

Time Exceeded Message. The IP protocol requires any IP module that finds the time-to-live parameter of an IP-datagram equal to zero to discard the datagram. In addition, the IP module may also send this message.

Parameter Problem Message. Any IP module may send this message if it discovers a problem with an IP-datagram. In addition, an IP module may also send this message for any problem not covered by any other ICMP message.

Source Quench Message. This message requests the source host to reduce the rate at which it is sending data. A gateway may send a source quench message if it drops an IP-datagram because it does not have enough buffer space available to store the datagram. In addition, the destination host may send a source quench message if datagrams are arriving to fast to be processed.

Redirect Message. This type of ICMP message is sent by a gateway to a host on the same network to change the host's routing tables.

Information Messages. There are three types of information messages. Each type consists of a request message as well as a corresponding reply message.

Echo Request and Reply. All gateways must implement this type of ICMP message (3:19). A host sends the Echo Request message to a gateway. When a gateway receives the Echo Request message, it sends an Echo Reply message to the host by reversing the source and destination addresses.

Time Stamp Request and Reply. This message is similar to the Echo message except the timestamp messages carry time stamps as data. The sender places a timestamp in the data portion as it transmits the datagram. The echoer adds two time stamps to the message. First, it adds a receive timestamp when it receives the message. Then, the echoer adds a transmit timestamp as it transmits the datagram.

Information Request and Reply. This type of message was developed to support self-configuring systems; however, the Reverse Address Resolution Protocol (RARP) is a better method (3:19).

Internet Traffic

There are three general classes of traffic; high-throughput traffic, low-delay traffic, and real-time traffic (13:811). High-throughput traffic is typically the result of file transfers. This type of traffic requires routes with excess capacity rather than minimum delay (13:811). On the other hand, low-delay traffic requires a route which minimizes delay. This type of traffic is typically generated

during the interactive use of computers; remote editing, database interaction and time sharing access are examples (25:1). Real-time traffic requires both high throughput and minimal delay. For example, the transmission of digitized speech requires that transmission delay be less than some threshold and that a constant flow of data be maintained (13:811).

Internet traffic is often described as bursty. Bursty is defined in two ways. Opderbeck defines bursty as traffic characterized "by a large peak to average line utilization ratio" and places bursty traffic in the low-delay class of traffic (13:811). Pawlita provides a similar definition of bursty traffic. He says

> bursty traffic on a given channel is characterized by a low utilization factor :
>
> $$\mu = \frac{\text{mean message length/mean message interarrival time}}{\text{channel transmission capacity}}$$
>
> (15:525)

Pawlita attributes bursty traffic to dialogue which results in alternating periods of high and low activity (15:525)

III. Congestion Control

Introduction

This chapter discusses the present method used by the Internet to control congestion and the two proposed methods. It begins with a brief discussion of the Source Quench method which is currently in use. Then, Nagle's Fair Queueing method is introduced. Finally, Zhang's Metered Queueing method is introduced.

Source Quench Method

Although all gateways are required to implement the procedure for sending source quench messages, it is recognized as an imperfect method for controlling Internet congestion (3:17). The source quench method has two inherent problems. First, sending source quench messages consumes bandwidth on the reverse channel and may contribute to congestion. Second, preparing and sending source quench messages consumes gateway CPU time. Both bandwidth and CPU time are critical resources to a congested network. For these reasons, when (and if) to send source quench messages is not specified. This decision is left to the implementation. Furthermore, how a host is expected to respond to the receipt of a source quench message is not specified. In fact, Zhang states "most host implementations ignore source quench messages even if they receive any" and concludes that "the source quench method is virtually non-existing "(31:1).

Nagle's Fair Queueing Method

Introduction. John Nagle introduced this method of congestion control while employed by the Ford Aerospace Communication Corporation. The classic approach to congestion control focuses on buffer management. By first assuming a packet switch with infinite storage, Nagle shows that congestion will still occur if the network is overloaded. Based on this observation, Nagle shows that network performance can be improved by departing from the traditional first-in-first-out (FIFO) method of transmitting packets.

This section begins with an brief examination of a typical packet switch. Next, it presents Nagle's argument that a packet switch with infinite storage will still become congested. Then, Nagle's method of Fair Queueing is discussed. Finally, a simple analysis of Nagle's Fair Queueing method is presented.

Packet Switch. As Figure 12 shows, a packet switch is a node with several incoming and several outgoing links, each of which is capable of transferring data at a specified rate. Packets arrive the switch on the incoming links. As they arrive, the switch reads the packet header to determine the address of the packet's destination. Using this address, the switch determines over which of its outgoing lines the packet should be sent. Then, the switch places the packet on the queue associated with the outgoing line (Figure 12). There, the packet waits its turn to be transmitted over the link.

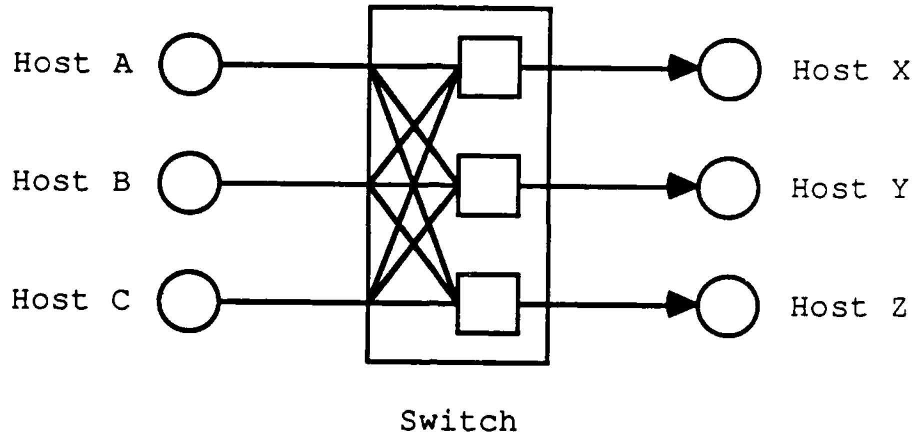


Figure 12. Packet Switch Node

Congestion with Infinite Storage. In all real implementations of the packet switch described above, the number of buffers available are limited. Therefore, the length of each of the outgoing queues is also limited. Classical methods of congestion control attempt to limit the flow of packets through the switch to a level which does not exhaust the storage available at the switch. This precludes a switch from having to discard a packet because it does not have room for it.

Nagle analyzes a generalized packet switch with infinite storage

space and shows congestion will still occur when the network is overloaded. In this analysis, he also assumes each packet has a finite life time, such as the Time-to-Live mechanism used in the Internet Protocol. Nagle shows that queue lengths will become so long that the amount of time a packet spends in the queue waiting to be transmitted will exceed its maximum lifetime. Thus, when the packet reaches the head of the queue its time to live value will be zero (or less) and the switch will have to discard it rather than transmit it. Thus, Nagle concludes "a datagram network with infinite storage, FIFO queueing, and a finite packet lifetime will, under overload, drop all packets" (12:3).

Nagle also shows the results of his analysis apply to networks with finite resources as well. The finite life time of a packet establishes an upper bound on the total storage required at a switch (12:2). This limit is fixed by the maximum value for the time to live parameter and the data rate of the incoming lines. Nagle uses the following example to demonstrate this effect.

> Consider a pure datagram switch for an ARPANET-like network. For the case of a packet switch with four 56kb links, and an upper bound on the time-to-live of 15 seconds, the maximum buffer space that could ever be required is 420K bytes. A switch provided with this rather modest amount of memory need never drop a packet due to buffer exhaustion (12:3).

Fair Queueing. Thus far, Nagle has shown that increasing the buffer space will not control congestion in a system such as the Internet. The solution Nagle proposes is based on the concept of fairness. As it pertains to packet switching networks, the concept of fairness implies that "each source host should be able to obtain an equal fraction of the resources of each packet switch" (12:7).

This objective can be met by changing the queueing structure and discipline used in a packet switch. Instead of a single FIFO queue associated with each output line, Nagle proposes multiple queues, one for each source, for each output line with each set of queues being serviced in a round-robin manner. Figure 13 illustrates the new queue structure for the packet switch shown in Figures 12.

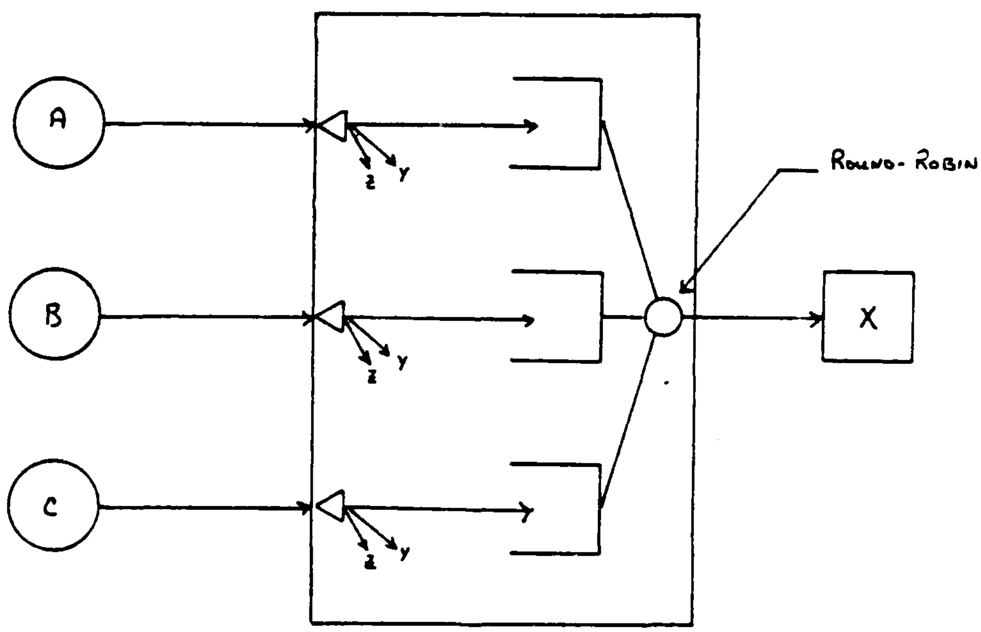


Figure 13. Fair Queueing Structure

Nagle's proposed queue structure dramatically changes the optimal strategy for a host. Under the present FIFO queueing discipline, a host can dominate the use of a link by sending packets as fast as it is able to. However, optimal strategy for a host under Nagle's Fair Queueing is to match the rate at which packets from its queue are removed from queue and transmitted. This means that the length of the queue associated with this host will be one. However, if the host chooses to exceed this rate, then the length of its queue in the packet switch gets longer. Thus, the only packets sent by this "badly-behaved" host experience an increase in delay time while packets sent by well-behaved hosts are not affected.

Analysis of Nagle's Fair Queueing Method. Using the simulation language SLAM, a simple simulation was performed to gain insight into the performance of Nagle's Fair Queueing method. Using the network model shown in Figure 14, simulations for both the present queue structure and Nagle's proposed queue structure were performed and the results compared.

Several assumption were made. First, packets were assumed to arrive at the switch according to a Poisson prooess. In addition, their size was assumed to be constant. Finally, the channel capacity was assumed to be 40 packets per second. Table III shows the arrival rates for each of the three cases simulated.

For each of the three cases, five runs were made with each run consisting of 1000 packets. The results of these five runs were averaged and are shown in Tables IV, V, and VI. These results confirm that Nagle's Fair Queueing method performs as predicted.

Table IV. Simulation Results: Time in System (seconds)

CASE	PRESENT	FAIR
1	0.046	0.046
2	0.220	0.219 A - 0.068 B - 0.341 C - 0.070
3	4.51	3.188 A - 0.0764 B - 5.30 C - 0.070

Table V. Simulation Results: Length of Queue and Wait Time

	Present		Fair					
			A		B		C	
CASE	L	W	L	W	L	W	L	W
1	0.5	0.02	0.16	0.02	0.17	0.02	0.17	0.02
2	7.32	0.19	0.35	0.04	6.5	0.31	0.4	0.05
3	214	3.75	0.4	0.05	213	5.2	0.4	0.04

L in packets W in seconds

Table VI. Simulation Results: Throughput (packets/second)

	PRESENT			FAIR		
CASE	A	B	C	A	B	C
1	8	8	8	8	8	8
2	8	20.2	8	8	20.2	8
3	5.6	28.5	5.8	8	23.9	8.2

	Arrival Rate (Packets/second)			
	Host A	Host B	Host C	Utilization
Case 1:	8	8	8	0.6
Case 2:	8	20	8	0.9
Case 3:	8	40	8	1.0

Zhang's Metered Queueing.

Introduction. Lixia Zhang proposed this method of controlling the congestion in the Internet in a draft paper she prepared while at the Laboratory for Computer Science at Massachusetts Institute of Technology. Zhang proposed a feedback control system designed to control the rate at which source hosts are allowed to send data through the Internet. Implementation of her method will require modification to the IP modules in the hosts and gateways and a modification of the ICMP Source Quench message. These modifications will allow the gateways to sense traffic levels and to tell the host when, by how much, and for how long they should reduce their traffic rates.

This section begins with a brief discussion of the assumptions Zhang made in developing her proposal. Then, the requirements she sought to satisfy are outlined. Next, the algorithm is presented. Finally, the changes required to implement Zhang's Metered Queueing in the host and gateways of the Internet are examined.

Assumptions. Zhang based the development of her proposed method for controlling Internet congestion on the following four assumptions:

1. Feedback congestion control is feasible.
2. The majority host-to-host data transmissions last, at least, one order of magnitude longer than their internet round-trip-time (RTT).
3. Gateways have adequate buffer spaces to save transient overflow traffic during the control response time period.

4. A single path is used at one time between a source-destination host pair (31:3).

Zhang's first assumption is fundamental to her algorithm while the next two assumptions are necessary to ensure the feedback control method is effective. The purpose of Zhang's last assumption is not made clear nor is it valid in the context of a system, such as the Internet, which provides datagram service. The second assumption allows sufficient time for the control message to reach the source host while the host is still transmitting the data that is causing the overload. Assumption three ensures there is sufficient storage capacity in the system to absorb the overload during the time it takes the control message to reach the source host and the time it takes for the effect of the control message to be felt at the congested gateway.

Requirements. Zhang established four general requirements the proposed congestion control fix must satisfy.

1. The fix "must last until the next generation of internet architecture" (31:3).
2. The resulting internet system must be as robust as the current system.
3. The fix must be capable of being implemented piecemeal.
4. The fix must be fair (31:3).

In addition to these general requirements, Zhang specified requirements which must be met at the hosts and gateways.

Host Requirements. There are two requirements which must be satisfied at the host. First, the control must affect all traffic (i.e., data, data retransmitted, and control) generated by the host. Second, the modification should be simple and introduce no overhead when there is no congestion (31:4).

Gateway Requirements. Zhang identified three requirements which have to be met at the gateways. First, the gateways must be able to assert control over the rate at which hosts are transmitting data. This implies the gateway should tell a host not only to reduce its rate, but also when to increase its rate again. Second, the gateway must have the means to enforce this control. Third, the fix must be "simple while flexible" (31:4). In addition to these three requirements, a gateway must also have the ability to sense congestion. Without this ability the gateway will not know when to tell a host to slow down nor when to tell the host it is OK to speed up again.

The Algorithm. Zhang's proposed algorithm works like this:

> Each gateway constantly observes its own traffic. When a congestion occurs, the gateway sends a revised ICMP source quench message to the responsible source hosts, informing them of how they should regulate their data transmissions. Each host must respond to the source quench message properly, otherwise its excessive packets may be discarded (31:4-5).

Changes. Zhang's algorithm requires the ICMP source quench message to carry two additional parameters. First, the revised source quench message will include a parameter which tells the host what rate it is permitted to transmit data at. The second parameter will tell the host

for how long it must reduce its rate. This parameter will depend on the "dynamic characteristics of the internet and internet traffic" and Zhang's paper does not describe or define this parameter any further (31:5). In addition to these changes to the ICMP source quench message, changes are required in the hosts and gateways.

Host Changes. A quench message table must be added to the host IP module. Each time the host receives a source quench message, it will check the quench message table to see if an entry exists for the destination address. If not, then the host will create an entry in the table for the source quench message. If an entry already exists, the host will update the quench message table with the information contained in the source quench message. The host will use the information contained in the table to control the rate at which it transmits data.

Gateway Changes. Two changes must be made to the IP module at each gateway. First, a control box which contains entries for every transmitting host must be added (31:6). The second change adds a data structure which is designed to order the packets waiting to be transmitted. This data structure is essentially an implementation of Nagle's Fair Queueing. The gateway uses the control box and data structure to determine the rate each host should transmit at and to decide when to send a source quench message.

Analysis of Zhang's Metered Queueing Method. Zhang's Metered queueing method is not complete. Two parameters, which are critical to the performance of the algorithm are not provided. The first parameter missing is the interval of time over which traffic should be averaged. This parameter is necessary for the gateways to predict congestion. The second missing parameter is the expiry time. This parameter specifies how long the control over the rate at which a host may send packets remains in effect. Both parameters rely heavily on the unknown nature of the internet traffic. Furthermore, determining the values of these parameters is clearly outside the scope of this research.

IV. Development of the Simulation Model

Introduction

The objective of this research is to determine whether Nagle's Fair Queueing or Zhang's Metered Queueing can control Internet congestion better than the Source Quench method presently in use. However, as discussed in the previous chapter, Zhang's Metered queueing method is not complete. Therefore, this thesis reduces to comparing Nagle's method with the Source Quench method. Thus, the hypothesis tested by this thesis is that the average delay a message traversing the Internet experiences when the Source Quench congestion control method is used is the same as the case when Nagle's Fair Queueing method is used. Testing this hypothesis required a total of three models; a model of the Internet traffic and two models of the Internet system. Both models of the Internet system are identical except that one implements the Source Quench method while the other implements Nagle's Fair Queueing. All models are implemented in the simulation language for alternative modeling, SLAM (24).

The next section outlines the general approach this researcher has taken toward modeling the Internet system. Then, the following sections explain each of the three models and the experimental procedures.

Internet Model

The Internet is a complex system composed of various heterogeneous networks interconnected by gateways. The protocol, IP, is common to all hosts and gateways connected to the Internet system. Thus, IP serves as the thread which ties the networks together. IP only requires datagram service from each network of the internet. Other than that, IP places no demands nor makes no assumptions about the service the networks provide.

In general, a network is comprised of three types of elements.

1. The hosts upon which the processes which require the services of the network reside.
2. The switches which route the data from one host to another.
3. The communication links which connect the switches together.

The Internet system can also be modelled in terms of these three basic elements. The hosts connected to the Internet are the same hosts connected to the constituent networks. However, in the Internet system, gateways perform the internet routing functions while the constituent networks become the the links which tie the gateways together. Each constituent network brings to the Internet system its characteristic performance parameters. Figure 15 uses this approach to represent that portion of the Internet required to establish communications between Hosts A, B, C, D and their common destination host. This figure forms the basis for the SLAM models discussed in the following sections.

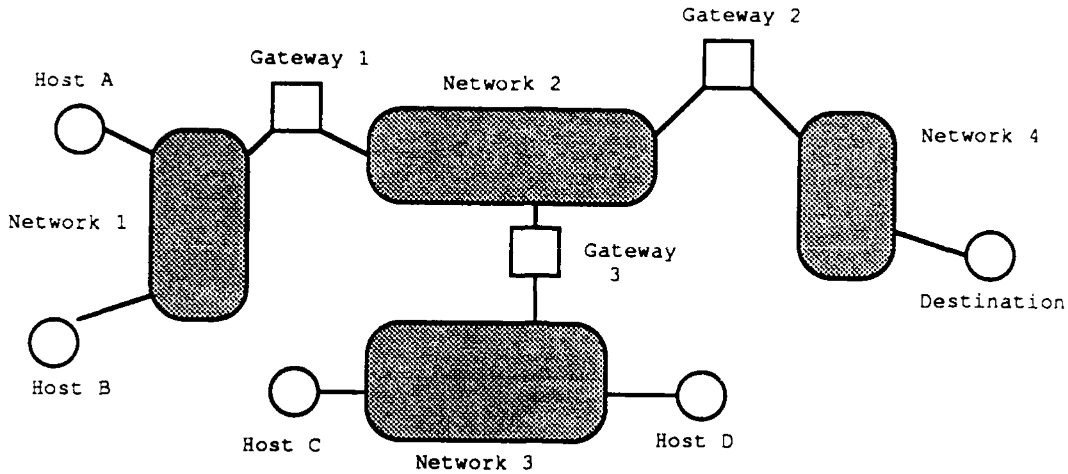


Figure 15. Internet Model

Internet Traffic Model

Actual traffic data is not available for analysis nor has the researcher found any references to previous analysis of the Internet traffic. Therefore, a model had to be developed which satisfied the descriptions of the Internet traffic. Internet traffic is frequently described as bursty. Chapter II described bursty as traffic which results in a high peak to average line utilization ratio. In addition, depending on its source, Internet traffic is either low or high in

volume. For example, processes using Telnet would result in low volume traffic while processes using FTP would normally result in high volume traffic, in one direction at least. Furthermore, early studies of the ARPANET showed that over 90% of the messages transmitted consisted of just one packet and that the average length of a message was only 243 bits (10:304-306).

Using this information, the researcher developed the following model:

Message Size = X + Y * I

where:

X is an exponentially distributed random variable with mean = 256 bits

Y is an exponentially distributed random variable with mean = 49,152 bits (6K bytes).

I is a binary random variable. Ninety of the time I = 0 and 10% of the time I = 1;

Internet with Source Quench

This section discusses the SLAM implementation of the Internet model with source quench congestion control. This model is based on the Internet system depicted in Figure 15. Since the Source Quench method of congestion control is virtually nonexistant in practice, it is not implemented in this model. That is, gateways do not send source quench

messages when they are forced to drop datagrams. The SLAM implementation of each of the three basic elements (host, gateway, and network) is discussed below.

Host. The four modules shown in Figure 16 represent the seven layers of the DoD Internet Protocol hierarchy. The application module includes the processes operating on the host and the utility layer protocol they are using to transfer data across the Internet. The TCP and IP modules reflect the various functions performed by the TCP protocol and the IP protocol respectively. Finally, the network interface module collectively represents the three lower layers (i.e., network, link, and physical layers). The following paragraphs explain the operation and implementation of each module in detail.

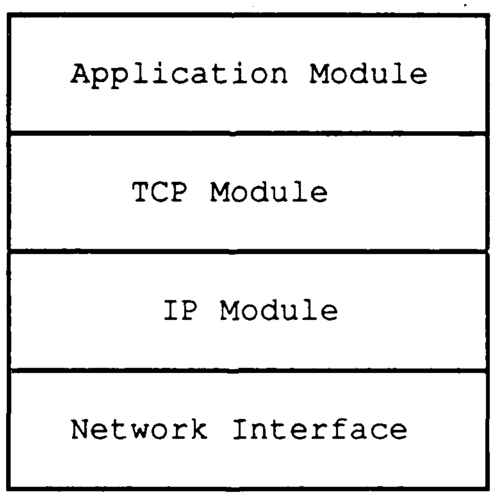


Figure 16. Internet Model - Host

Application Module. This module creates SLAM entities with exponentially distributed interarrival times. At the time of their creation, each entity represents a message. In addition, each entity has eight attributes defined in Table VII. The Application module assigns values to the first 4 attributes. The size of the message is determined by the Internet traffic model described earlier. However, instead of assigning the size of a message in bits, the size is given in blocks of 512 bytes. Thus, the size of the message represents the number of TCP segments in the message.

Table VII. SLAM Attributes

Attribute	Definition
1	Time of Creation
2	Source of Message
3	Size of Message in 512 byte segments
4	Message ID number
5	NATRS used by the batch node
6	Retransmit Timer
7	Time to Live parameter
8	Time entered gateway

TCP Module. The TCP module performs three separate functions. First, the TCP module calculates the time interval for the retransmission timer and places it in attribute 6 of the message. The modules uses the global variable Beta and its smoothed round trip time (SRTT) to calculate the retransmit interval according to:

Retransmit Interval = Beta * SRTT

Each TCP module updates its SRTT each time a message is received using:

SRTT = Alpha * SRTT + (1 - Alpha) * RTT

where:

Alpha = 0.9 (used by 4.3 BSD Unix)

Beta = 2.0 (used by Unix 4.3 BSD)

RTT   = 2 * (message's time in system)

The second function the TCP module performs is to divide the message into 512 byte segments. The SLAM unbatch statement is used for this purpose. As a message passes through the unbatch node it is replicated according to its size. For example, if attribute 3 of the message is 5, then five identical segments will leave the node. Implicit in this process is the addition of a 20 byte TCP header to each

segment. In addition, each message is assumed to consist of an integral number of segments.

The destination TCP is required to deliver messages to it's user in the same form as the source TCP received the message from it's user. Therefore, the messages which were divided into segments at their source must be reassembled at the destination. This is the third function of the TCP module. The SLAM batch statement performs this operation. As segments arrive the destination, they are sorted according to their source. Then the segments are routed to a batch node. Here the segments are batched according to their message ID (attribute 4). Messages are released only after the number of segments contained in attribute 3 have been received. Message service time statistics are collected at this point.

<u>IP Module</u>. The IP module receives the segments from the TCP module. The IP module forms a datagram from the segment by setting the time to live (TTL) attribute to 15 seconds. In addition, a 20 byte IP header is also assumed to be added to the datagram. After it has finished processing the datagram, the IP module places the datagram in a queue where it waits for the network interface module.

<u>Network Interface Module</u>. The network interface module removes datagrams from its queue and transmits them one at a time. The time it takes to transmit a datagram is assumed to be exponentially distributed. The mean transmission time is determined by the data rate

of the line the host is connected to and the size of the datagram. For the purpose of this experiment, all lines are assumed to be 56k bps lines which are standard for ARPANET like networks. The average size of the datagram is assumed to be 72 bytes. This figure accounts for the TCP header, IP header, and allows for 32 bytes (256 bits) of data which is consistent with the traffic model. Thus, the mean transmission time is 10.3 m sec. In comparison, if all the datagrams were assumed to be the maximum size, then they would take 78.9 m sec to transmit. Once the network interface has transmitted it, the datagram is on a network.

Network. There are three networks in this model. Each network is represented by an exponentially distributed delay. A mean delay time of 90 m sec is used for all three networks. There is no limit on the number of datagrams that can be on the network at any one time. However, the network interface modules control the rate at which datagrams enter the networks.

Gateway. The gateways consists of two modules; an IP module and a network interface module.

IP Module. The gateway IP module performs two functions. First it implements the IP TTL function. The IP specifies that the TTL parameter of a datagram be reduced by one each time the datagram is processed by an IP module. Therefore, as datagrams arrive at the gateway IP module, their TTL parameter is reduced by 1. In addition,

the time the datagram arrives at the gateway is also recorded. Then, just before a datagram departs the gateway, its TTL parameter is reduced by the amount of time the datagram remained in the gateway. After reducing the datagram's TTL parameter, the IP module checks the value of the TTL parameter. Datagrams with TTL parameters less than or equal to zero are not transmitted; they are discarded instead. The second function the gateway IP module performs is queue management. Datagrams which arrive at the gateway when the queue is full are also discarded.

Discarded datagrams are returned to their source for retransmission. However, their return is delayed by the amount of time remaining on their retransmit timer. For example, suppose the retransmit timer of a dropped datagram was set to 600m sec and the datagram had been in the system for 120 m sec. Then it would take this datagram another 380 m sec to reach its source host. This procedure models the retransmission function; however, it excludes the possibility of duplicate datagrams being in the system.

Network Interface Module. The gateway network interface module is the same as the host interface module and it uses the same parameters in this experiment.

Internet with Nagle's Fair Queueing

The model for the Internet with Nagle's Fair Queueing is exactly the same as the Internet Model for the Source Quench method except for

the queue structure of the queues of the gateways. In this model, the single queue per output line is replaced by multiple queues; one for each source. In order to maintain the storage capacity of the gateway at the same level, the maximum length of each of the multiple queues has been reduced by 1/2 where two queues have replaced one and by 1/4 where four queues replaced one.

Experimental Procedure

The experimental portion of this research passed through five phases. The first phase verified that the models operated as intended. During this phase, extensive use of the SLAM trace option allowed the researcher to trace messages through the system. This procedure ensured the models performed properly.

Next, a set of pilot runs were made. Using these pilot runs, the lower and upper bounds of the operating range were determined. In addition, the pilot runs established the duration of a run and the number of runs required.

Then, a complete set of runs were made using the source quench model. During these runs, the message arrival rates of each host was varied from 1.0 to 20 messages per second. Figure 17 shows the delay curve produced by these runs. The curve is typical of a computer network and serves to validate the simulation model. The data collected during these runs was analyzed. From this analysis, a message arrival rate which resulted in a network utilization of approximately 60%. This

rate, 7.5 messages per second, established the operating level of a well-behaved host.

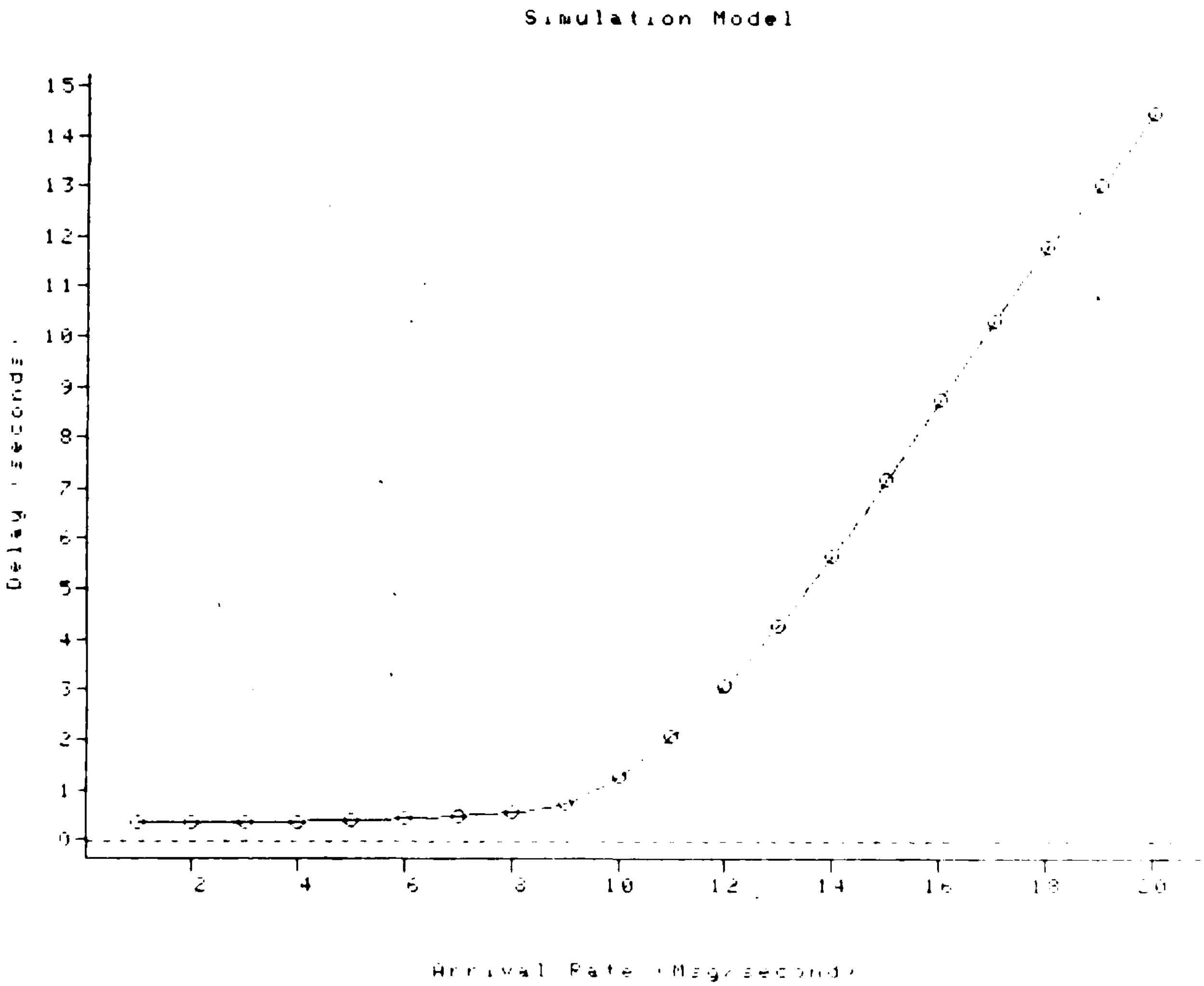


Figure 17. Delay Curve of the Simulation Model

During the next phase of the experiment, the message generation rates of Hosts A, C, and D were set to the rate of a well-behaved host while the message generation rate of Host B was varied from 1.0 to 20.0 messages per second. Data was collected during each of the runs. Next, the delay data obtained during these runs was averaged and plotted against the message arrival rate. This procedure was repeated with Nagle's Fair Queueing model. Then, the results of the two experiments were compared using paired difference test to determine whether the delay times were equal or not. The next chapter presents these results.

V. Simulation Results

Introduction

This chapter presents the results of the simulations. The first section presents the results of the simulation of the Internet model. The second section presents the simulation results with Nagle's Fair queueing implemented in the gateways. Then, the third section compares the results of the two simulations. Two methods of comparison are used, First, the two models are compared on the basis of their delay curves. Then, a Paired Difference test is used to compare the delay data gathered during the simulation. This delay data, which was extracted from the *SLAM Summary Reports*, is included as the Appendix.

Internet Model

Figure 18 shows the delay curve for this model. During the experiment, the mean message arrival rates for Hosts A, C, and D were held at 7.5 messages/second while the mean message arrival rate for Host B was varied from 1 to 20. Thus, the horizontal axis in Figure 18 represents the arrival rate for Host B. This graph clearly shows the effect increasing the message arrival rate at Host B has on the delay experienced by messages generated at other locations within the Internet. The tables included in the Appendix show that this effect is

similar for messages generated at Host C and D; however, in order to keep the figures simple, only the data pertaining to Host A and B are plotted.

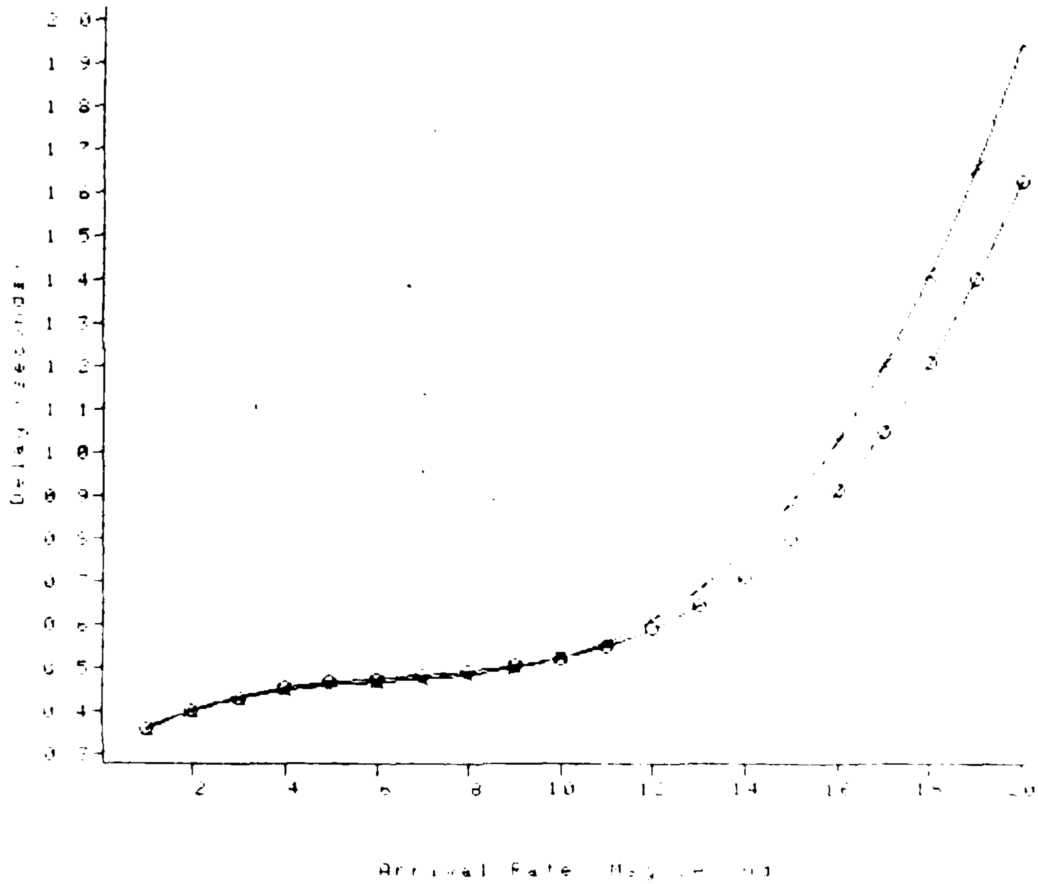


Figure 18. Delay Curve - Internet Model

Nagle's Fair Queueing Model

Figure 19 shows the delay curve for the Internet model with Nagle's Fair queueing implemented. The same procedures were used to generate this curve. However, using Nagle's method, the effect Host B has over the delay of messages generated by other hosts in the system is greatly reduced. In addition, the delay experienced by messages sent by Host B increases sharply once Host B exceeds the rate of approximately 8.0 message/second. This is exactly the effect Nagle predicted.

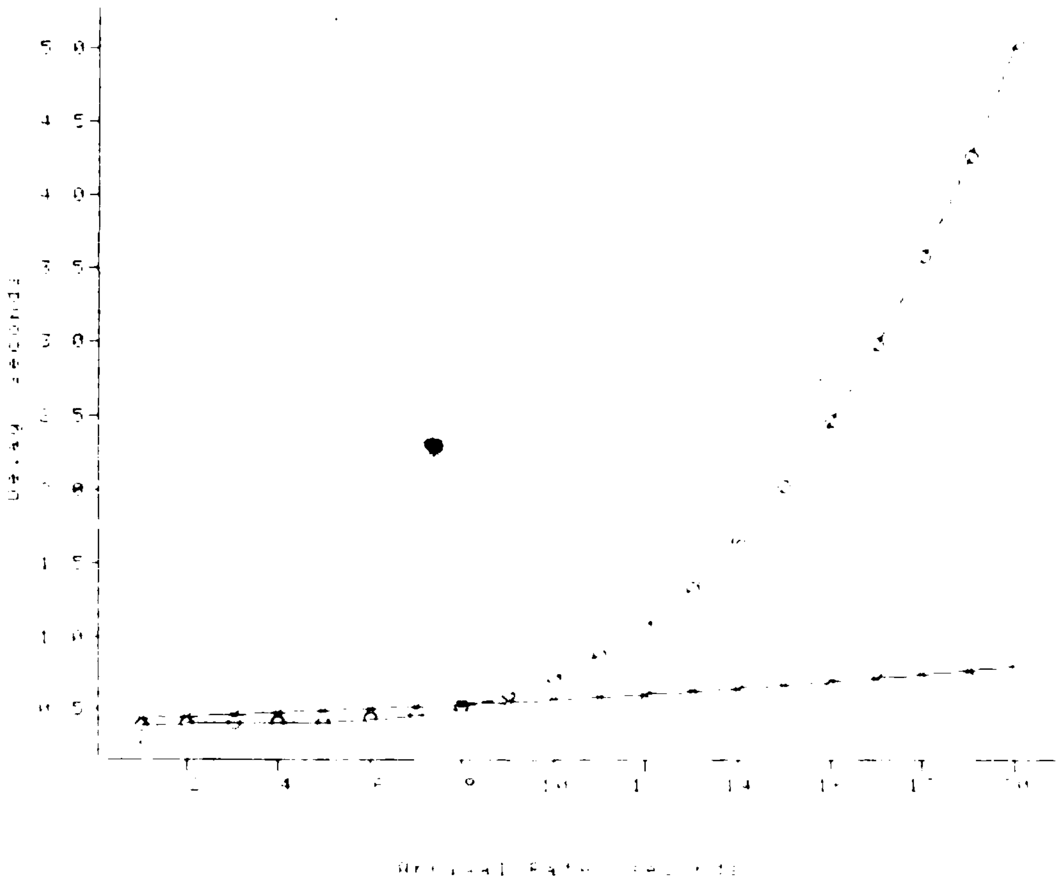


Figure 19. Delay Curve - Nagle's Model

Comparison

Delay Curves. Figure 20 compares the delay curves for messages sent by Host A in each of the two model. There are two points of interest. First, at slow arrival rates the Nagle's method introduces additional delay. This delay is undesirable and represents the overhead incurred by Nagle's method. However, this overhead can be explained by the buffer management scheme the model uses in its implementation of Nagle's method.

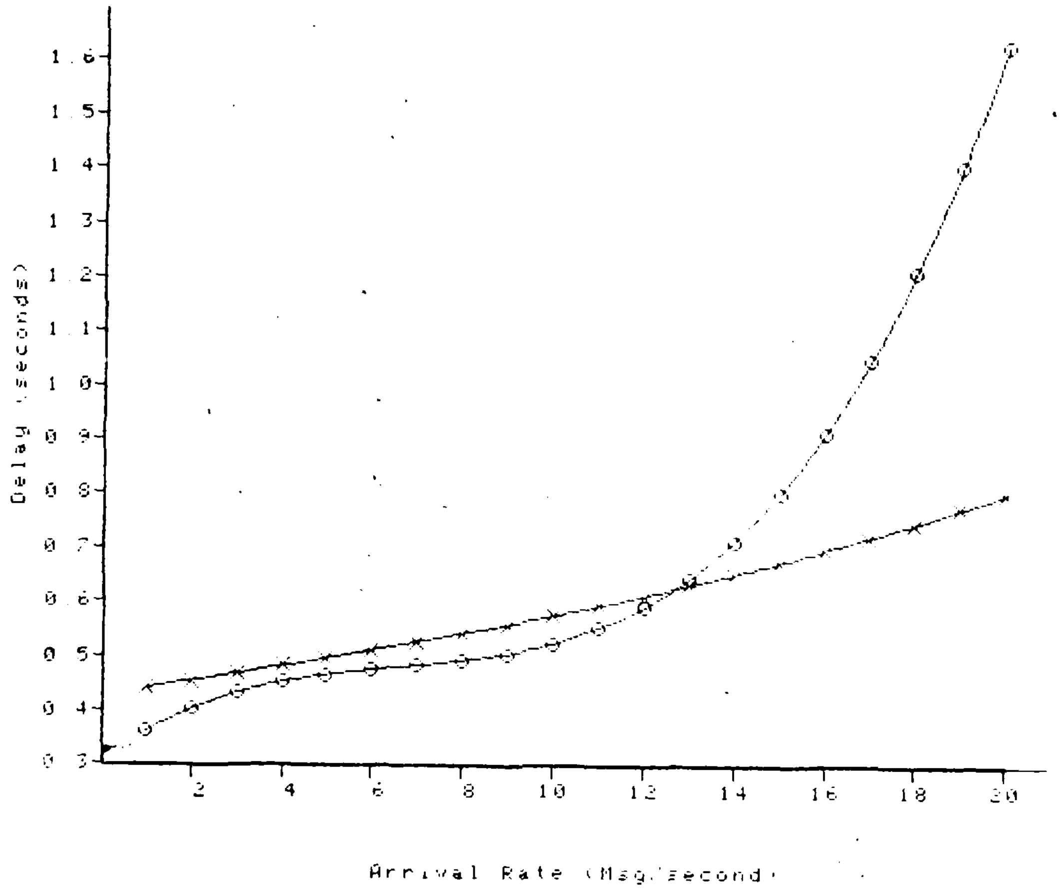


Figure 20. Delay Curve - Host A Messages

In the Internet model, all gateways had maximum queue lengths of 16 datagrams. However, in Nagle's model, the single queues of the gateways were divided into multiple queues, one for each host with traffic passing through the gateway. Therefore, the gateways shared by two host now had two queues with a maximum length of 8 datagrams each. Similarly, the gateway shared by all four hosts contained four queues, each with a maximum length of 4. In addition, no sharing of buffer resources took place in the model. That is, the queues of each gateway were for the exclusive use of a specific host. If they were not being used by that host, then they were wasted. As a result, the gateways were forced to drop more datagrams than before; although, the average length of the queues belonging to the well-behaved host averaged less than 1 datagram (i.e, buffer utilization was about 25% with queues of maximum length of 4 and 12.5% when the maximum length was 8). Therefore, a method of managing the gateway buffers which allows some sharing of buffer resources would reduce the overhead incurred in this model.

The second point of interest is the difference in the delays produced by the two models at the higher arrival rates. For example, when Host B is sending messages at the rate of 20 msg/second, Nagle's method reduces the delay a message sent by Host A experiences from 1.6 to 0.8 seconds. In addition, the rate of increase in Nagle's model is linear throughout the range of the simulation, while the rate of increase in the Source Quench model is very non-linear at the higher message arrival rates.

Figure 21 shows the delay curves for messages generated by Host B in the two models. This figure illustrates the punishing effect Nagle's method has upon badly-behaved hosts. For example, if a host chooses to send messages at a rate of 16 messages per second rather than at 8; its messages will be delayed 5 times as long. Furthermore, this delay is primarily the result of retransmissions which place an additional load on the host.

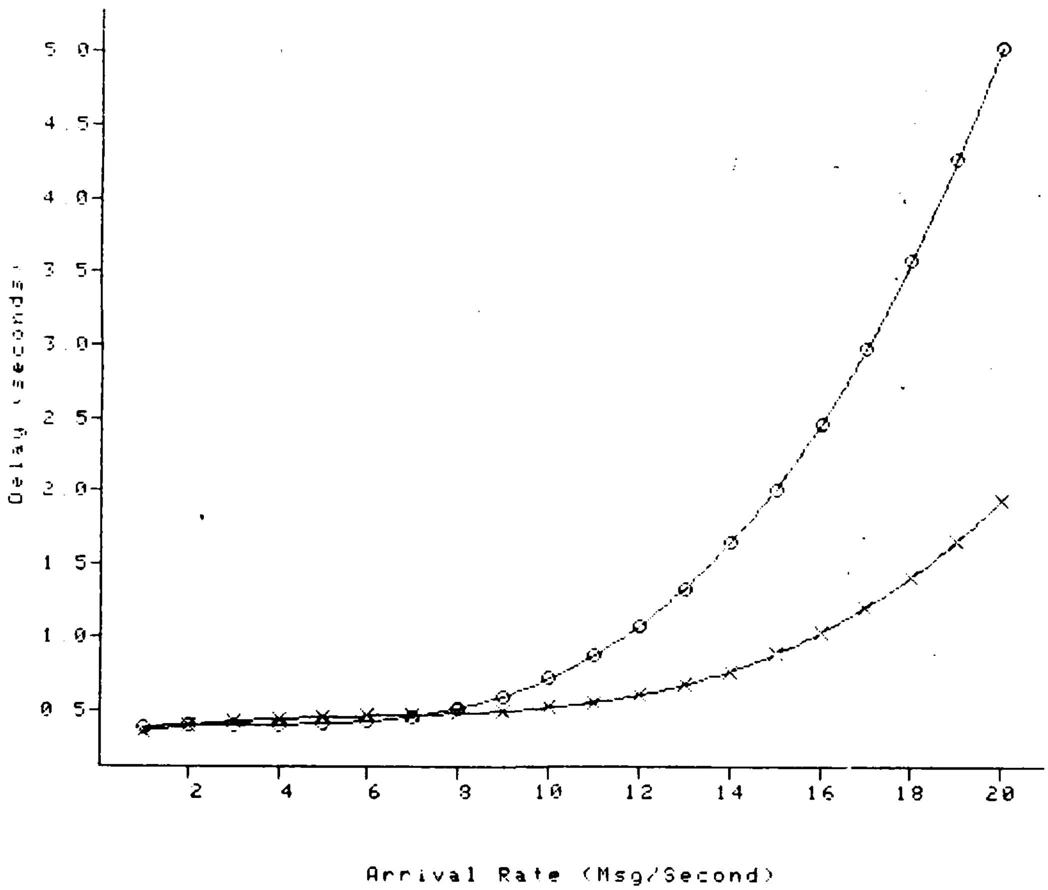


Figure 21. Delay Curve - Host B Messages

Paired Difference Test. Table VIII provides the results of the Paired Difference test. SAS, a software system for data analysis was used to perform this test (26). The test was conducted at a 95% significance level. The 'Yes' column of Table VIII, indicates that the null hypothesis should be rejected. Rejecting the null hypothesis means there sufficient evidence to conclude that there is a difference in the mean message delay times between the two models at that rate. Therefore, this column also indicates which of the two models had the greater delay rates.

The Paired Difference test confirms that Nagle's method, as implemented in the model, introduces overhead in terms of increased delay at slow arrival rates. For the well behaved hosts (Host A, B, and C), the table does not indicate a significant difference in the mean delay times between the two models until Host B begins sending messages at the rate of 19 messages per second. However, the test does indicate a significant difference in the mean delay time for message sent by Host B at rates greater than 11 messages per second.

Table VIII. Paired Difference Test Results

Rate	Host A Yes	Host A No	Host B Yes	Host B No	Host C Yes	Host C No	Host D Yes	Host D No
1.0	3>2			X		X	3>2	
2.0	3>2		2>3		3>2			X
3.0	3>2			X	3>2			X
4.0		X		X		X	3>2	
5.0		X		X		X		X
6.0	3>2		3>2		3>2			X
7.0	3>2			X	3>2		3>2	
7.5		X		X		X		X
8.0		X		X		X		X
9.0		X		X		X		X
10.0	2>3			X		X		X
11.0		X	3>2			X		X
12.0		X	3>2			X	3>2	
13.0		X	3>2			X		X
14.0		X	3>2			X		X
15.0		X	3>2			X		X
16.0	2>3		3>2			X		X
17.0		X	3>2			X		X
18.0		X	3>2			X		X
19.0	2>3		3>2		2>3		2>3	
20.0	2>3		3>2		2>3		2>3	

VI. Conclusions and Recommendations

Introduction

This chapter presents the conclusions and recommendations. The first section presents the conclusions. These conclusions are based on the results of this research project. The second section presents several recommendations for future study.

Conclusions

This thesis has shown that Nagle's Fair Queueing has the potential to improve the performance of the Internet. In particular, this thesis has shown that Nagle's Fair queueing to be effective at preventing a badly behaved host, or any source of network traffic, from dominating the capacity of the network. These results justify the development and implementation of a project to test Nagle's Fair Queueing in several Internet gateways for the purpose of determining whether it should be implemented throughout the system.

Recommendations

During the course of this research effort, several areas were uncovered which require further study. The purpose of this section is to outline those areas. The first pertains to Internet traffic data -- none exists. It is very difficult to solve a problem before it is

defined. Is there congestion at the gateways? If so, is the congestion do to limited CPU capacity or is it the result of line capacity. Is the traffic bursty, for that matter is it truly random? These are a few of the questions that need to be answered. To answer them requires careful planning. This plan could be developed by a graduate student interested networks and evaluation of their performance. Implementation of such a plan would have to be done through the program management office. However, analysis of the data obtained through the study could provide a second thesis in the future.

A second area open to study pertains to the response a host should take when it learns it may be responsible for or is contributing to congestion in the Internet. Zhang's Metered Queueing algorithm is one approach and it should be developed further. Prue recently introduced Squid as a second. Both methods attempt to predict congestion, inform the host, and define a host's response.

A third area pertains to performing simulation in Ada. Such a research project could begin with defining what sort of tools are necessary for simulation. Then, proceed to uncover and evaluate those available in the Ada language. Finally, document the findings by describing what is available and what needs to be developed.

A final area of recommended study pertains to the protocols TCP and IP. Each of these protocols requires a header of at least 20 bytes. In addition, TCP requires the sending of several empty segments (headers alone) to open and close connections. What effect does this overhead have on network performance?

Appendix: ANALYSIS OF SLAM DATA.
HOST A

MRATE=0.5

OBS	MOD2DEL	MOD3DEL	DIFF
1	0.3667	0.4341	0.0674
2	0.4000	0.4415	0.0415
3	0.3818	0.4368	0.0550
4	0.4213	0.4225	0.0012
5	0.3856	0.4481	0.0625
6	0.3700	0.3784	0.0084
7	0.3921	0.5585	0.1664
8	0.3875	0.4341	0.0466
9	0.3987	0.3964	-0.0023
10	0.3860	0.4914	0.1054

MRATE=1

OBS	MOD2DEL	MOD3DEL	DIFF
11	0.4105	0.5175	0.1070
12	0.4027	0.4539	0.0512
13	0.4109	0.4212	0.0103
14	0.4043	0.5393	0.1350
15	0.4077	0.4679	0.0602
16	0.3973	0.5012	0.1039
17	0.3994	0.4621	0.0627
18	0.3671	0.4070	0.0399
19	0.4076	0.4033	-0.0043
20	0.3835	0.5244	0.1409

MRATE=2

OBS	MOD2DEL	MOD3DEL	DIFF
21	0.3952	0.5405	0.1453
22	0.4264	0.4520	0.0256
23	0.4411	0.4474	0.0063
24	0.4095	0.4069	-0.0026
25	0.3938	0.4391	0.0453
26	0.3970	0.4134	0.0164
27	0.4172	0.4405	0.0233
28	0.3940	0.4238	0.0298
29	0.3935	0.4716	0.0781
30	0.4032	0.4051	0.0019

Appendix: ANALYSIS OF SLAM DATA.
HOST A

------------------------------ MRATE=3 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
31	0.4418	0.4612	0.0194
32	0.4072	0.3965	-0.0107
33	0.3881	0.4331	0.0450
34	0.4004	0.4950	0.0946
35	0.4152	0.4370	0.0218
36	0.4076	0.4225	0.0149
37	0.4215	0.4496	0.0281
38	0.4191	0.4232	0.0041
39	0.4594	0.5672	0.1078
40	0.3959	0.4031	0.0072

------------------------------ MRATE=4 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
41	0.3952	0.4333	0.0381
42	0.5016	0.4407	-0.0609
43	0.4873	0.4413	-0.0460
44	0.4117	0.4866	0.0749
45	0.4132	0.4773	0.0641
46	0.4107	0.4357	0.0250
47	0.4234	0.4221	-0.0013
48	0.3906	0.5018	0.1112
49	0.4731	0.4466	-0.0265
50	0.4677	0.4940	0.0263

------------------------------ MRATE=5 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
51	0.4260	0.4260	0.0000
52	0.4783	0.8693	0.3910
53	0.4412	0.5210	0.0798
54	0.4391	0.5442	0.1051
55	0.4330	0.4168	-0.0162
56	0.4509	0.5067	0.0558
57	0.4114	0.4838	0.0724
58	0.4548	0.4687	0.0139
59	0.4087	0.4735	0.0648
60	0.4844	0.4987	0.0143

Appendix: ANALYSIS OF SLAM DATA. HOST A

MRATE=6

OBS	MOD2DEL	MOD3DEL	DIFF
61	0.3843	0.4912	0.1069
62	0.3780	0.5598	0.1818
63	0.4163	0.4446	0.0283
64	0.4679	0.5022	0.0343
65	0.4117	0.6130	0.2013
66	0.4058	0.4029	-0.0029
67	0.3956	0.5451	0.1495
68	0.4441	0.4845	0.0404
69	0.4154	0.6544	0.2390
70	0.4361	0.4650	0.0289

MRATE=7

OBS	MOD2DEL	MOD3DEL	DIFF
71	0.4385	0.5902	0.1517
72	0.4338	0.5120	0.0782
73	0.4551	0.5437	0.0886
74	0.5274	0.5196	-0.0078
75	0.4614	0.5364	0.0750
76	0.4769	0.5240	0.0471
77	0.4853	0.5041	0.0188
78	0.5093	0.5187	0.0094
79	0.5289	0.5259	-0.0030
80	0.4244	0.4792	0.0548

MRATE=7.5

OBS	MOD2DEL	MOD3DEL	DIFF
81	0.5523	0.4607	-0.0916
82	0.4431	0.6722	0.2291
83	0.4669	0.6784	0.2115
84	0.4969	0.5490	0.0521
85	0.4201	0.4625	0.0424
86	0.5024	0.4589	-0.0435
87	0.5899	0.5192	-0.0707
88	0.7042	0.5361	-0.1681
89	0.4656	0.4540	-0.0116
90	0.4970	0.5601	0.0631

Appendix: ANALYSIS OF SLAM DATA.
HOST A

MRATE=8

OBS	MOD2DEL	MOD3DEL	DIFF
91	0.5151	0.4624	-0.0527
92	0.5921	0.5089	-0.0832
93	0.4868	0.5475	0.0607
94	0.5112	0.4221	-0.0891
95	0.4368	0.5848	0.1480
96	0.6163	0.5048	-0.1115
97	0.5053	0.5725	0.0672
98	0.4669	0.5426	0.0757
99	0.4925	0.4753	-0.0172
100	0.4253	0.3983	-0.0270

MRATE=9

OBS	MOD2DEL	MOD3DEL	DIFF
101	0.4796	0.5073	0.0277
102	0.5075	0.5306	0.0231
103	0.4722	0.4824	0.0102
104	0.6137	0.4939	-0.1198
105	0.5233	0.6174	0.0941
106	0.4589	0.5296	0.0707
107	0.6200	0.5859	-0.0341
108	0.5397	0.7224	0.1827
109	0.4876	0.4511	-0.0365
110	0.4768	0.5977	0.1209

MRATE=10

OBS	MOD2DEL	MOD3DEL	DIFF
111	0.5231	0.5476	0.0245
112	0.5856	0.4339	-0.1517
113	0.5146	0.6184	0.1038
114	0.6989	0.6585	-0.0404
115	0.5801	0.4541	-0.1260
116	0.6859	0.5693	-0.1166
117	0.6356	0.4641	-0.1715
118	0.6510	0.5497	-0.1013
119	0.5942	0.5756	-0.0186
120	0.4642	0.4233	-0.0409

Appendix: ANALYSIS OF SLAM DATA.
HOST A

----------------------------- MRATE=11 -----------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
121	0.4796	0.4751	-0.0045
122	0.5075	1.0840	0.5765
123	0.4722	0.5804	0.1082
124	0.6137	0.7522	0.1385
125	0.5233	0.5380	0.0147
126	0.4589	0.5300	0.0711
127	0.6200	0.5641	-0.0559
128	0.5397	0.5606	0.0209
129	0.4876	0.5930	0.1054
130	0.4768	0.5910	0.1142

----------------------------- MRATE=12 -----------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
131	0.5606	0.4995	-0.0611
132	0.5321	0.6760	0.1439
133	0.5858	0.9094	0.3236
134	0.6363	0.4452	-0.1911
135	0.6337	0.5632	-0.0705
136	0.4899	0.6436	0.1537
137	0.7887	0.8603	0.0716
138	0.4988	0.7835	0.2847
139	0.5878	0.5525	-0.0353
140	0.5370	0.9875	0.4505

----------------------------- MRATE=13 -----------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
141	0.5630	0.5722	0.0092
142	0.7673	0.8132	0.0459
143	0.6048	0.4457	-0.1591
144	0.6301	0.5904	-0.0397
145	0.5370	1.1270	0.5900
146	0.6360	0.6766	0.0406
147	0.6602	0.5820	-0.0782
148	1.0470	0.6377	-0.4093
149	0.5137	0.4861	-0.0276
150	0.7445	0.8139	0.0694

Appendix: ANALYSIS OF SLAM DATA.
HOST A

------------------------------ MRATE=14 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
151	0.8166	0.7485	-0.0681
152	1.1815	0.6919	-0.4896
153	0.7361	0.7050	-0.0311
154	0.7045	0.6776	-0.0269
155	1.1139	0.5590	-0.5549
156	0.5523	0.6464	0.0941
157	0.6866	0.4994	-0.1872
158	0.5004	0.7028	0.2024
159	0.6536	0.4828	-0.1708
160	0.7652	0.5059	-0.2593

------------------------------ MRATE=15 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
161	0.8070	0.6833	-0.1237
162	0.8102	0.7777	-0.0325
163	0.8448	0.5513	-0.2935
164	0.7612	0.5887	-0.1725
165	1.0500	0.8298	-0.2202
166	0.6733	0.6438	-0.0295
167	1.4260	0.5487	-0.8773
168	0.5828	0.6780	0.0952
169	0.7310	0.6923	-0.0387
170	0.6641	0.8270	0.1629

------------------------------ MRATE=16 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
171	0.7887	0.6187	-0.1700
172	0.7627	0.5879	-0.1748
173	1.1144	0.5941	-0.5203
174	0.9064	0.6459	-0.2605
175	1.1001	0.4408	-0.6593
176	1.1265	0.6440	-0.4825
177	0.7442	0.7802	0.0360
178	0.8247	0.6060	-0.2187
179	1.1139	0.7628	-0.3511
180	1.1058	0.7445	-0.3613

Appendix: ANALYSIS OF SLAM DATA.
HOST A

---------------------------- MRATE=17 ----------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
181	0.7262	0.6413	-0.0849
182	0.7319	0.6161	-0.1158
183	2.6110	1.0770	-1.5340
184	0.7512	0.7167	-0.0345
185	1.0280	0.5326	-0.4954
186	0.7695	0.9045	0.1350
187	1.0620	0.6114	-0.4506
188	0.6877	0.7198	0.0321
189	1.1030	0.5890	-0.5140
190	0.7424	0.6526	-0.0898

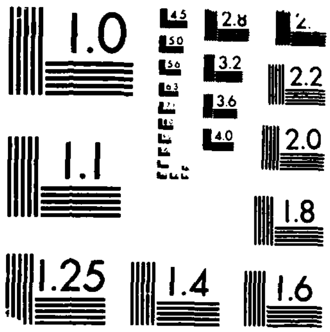
---------------------------- MRATE=18 ----------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
191	1.0620	0.8035	-0.2585
192	0.6872	1.2980	0.6108
193	0.9135	0.7279	-0.1856
194	0.6017	0.5197	-0.0820
195	0.6880	0.6083	-0.0797
196	1.3560	0.6650	-0.6910
197	0.8608	0.6383	-0.2225
198	0.7722	0.7977	0.0255
199	0.8191	0.6115	-0.2076
200	1.8760	0.8425	-1.0335

---------------------------- MRATE=19 ----------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
201	1.4970	0.6176	-0.8794
202	1.6060	0.6635	-0.9425
203	0.7997	0.5312	-0.2685
204	1.0400	0.6011	[illegible]
205	1.1710	0.6135	[illegible]
206	2.6070	1.0510	[illegible]
207	1.0240	0.686[illegible]	[illegible]
208	1.3770	[illegible]	[illegible]
209	1.3820	[illegible]	[illegible]
210	1.356[illegible]	[illegible]	[illegible]

AD-A190 574 EFFECTIVELY CONTROLLING DATAGRAM CONGESTION ON THE DOD 2/2
INTERNET SYSTEM GATEWAYS(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI B J SCHOFIELD
UNCLASSIFIED DEC 87 AFIT/GE/ENG/87D-57 F/G 25/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Appendix: ANALYSIS OF SLAM DATA.
HOST A

MRATE=20

OBS	MOD2DEL	MOD3DEL	DIFF
211	0.8829	1.5230	0.6401
212	1.1200	0.6497	-0.4703
213	3.5910	0.3775	-3.2135
214	1.7560	0.5999	-1.1561
215	1.3490	0.7487	-0.6003
216	1.9390	0.9538	-0.9852
217	1.4360	1.1660	-0.2700
218	1.9680	0.7426	-1.2254
219	1.9460	0.8079	-1.1381
220	1.7610	1.1550	-0.6060

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST A

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR¦T¦
MRATE=0.5				
DIFF	0.05521000	0.01624869	3.40	0.0079
MRATE=1				
DIFF	0.07068000	0.01572640	4.49	0.0015
MRATE=2				
DIFF	0.03694000	0.01415135	2.61	0.0283
MRATE=3				
DIFF	0.03322000	0.01229529	2.70	0.0243
MRATE=4				
DIFF	0.02049000	0.01737908	1.18	0.2686
MRATE=5				
DIFF	0.07809000	0.03689763	2.12	0.0634
MRATE=6				
DIFF	0.10075000	0.02736976	3.68	0.0051
MRATE=7				
DIFF	0.05128000	0.01564885	3.28	0.0096
MRATE=7.5				
DIFF	0.02127000	0.04011957	0.53	0.6088
MRATE=8				
DIFF	-0.00291000	0.02725305	-0.11	0.9173
MRATE=9				

DIFF	0.03390000	0.02761025	1.23	0.2507

------------------------- MRATE=10 -------------------------

DIFF	-0.06387000	0.02716007	-2.35	0.0432

------------------------- MRATE=11 -------------------------

DIFF	0.10891000	0.05554096	1.96	0.0815

PAIRED-DIFFERENCE TEST

MESSAGE DELAY BY RATE
HOST A

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>|T|
		MRATE=12		
DIFF	0.10700000	0.06413159	1.67	0.1296
		MRATE=13		
DIFF	0.00412000	0.07873153	0.05	0.9594
		MRATE=14		
DIFF	-0.14914000	0.07548339	-1.98	0.0796
		MRATE=15		
DIFF	-0.15298000	0.09155074	-1.67	0.1291
		MRATE=16		
DIFF	-0.31625000	0.06397083	-4.94	0.0008
		MRATE=17		
DIFF	-0.31519000	0.15353224	-2.05	0.0703
		MRATE=18		
DIFF	-0.21241000	0.13642661	-1.56	0.1539
		MRATE=19		
DIFF	-0.68294000	0.11969720	-5.71	0.0003
		MRATE=20		
DIFF	-0.90248000	0.31122250	-2.90	0.0176

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST A

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=0.5					
MOD3DEL	10	0.44418000	0.05017633	0.37840000	0.55850000
MOD2DEL	10	0.38897000	0.01563799	0.36670000	0.42130000
MRATE=1					
MOD3DEL	10	0.46978000	0.04957196	0.40330000	0.53930000
MOD2DEL	10	0.39910000	0.01384879	0.36710000	0.41090000
MRATE=2					
MOD3DEL	10	0.44403000	0.04002755	0.40510000	0.54050000
MOD2DEL	10	0.40709000	0.01642677	0.39350000	0.44110000
MRATE=3					
MOD3DEL	10	0.44884000	0.05040276	0.39650000	0.56720000
MOD2DEL	10	0.41562000	0.02150461	0.38810000	0.45940000
MRATE=4					
MOD3DEL	10	0.45794000	0.02888518	0.42210000	0.50180000
MOD2DEL	10	0.43745000	0.04072988	0.39060000	0.50160000
MRATE=5					
MOD3DEL	10	0.52087000	0.12858666	0.41680000	0.86930000
MOD2DEL	10	0.44278000	0.02526244	0.40870000	0.48440000
MRATE=6					
MOD3DEL	10	0.51627000	0.07718404	0.40290000	0.65440000
MOD2DEL	10	0.41552000	0.02763443	0.37800000	0.46790000
MRATE=7					
MOD3DEL	10	0.52538000	0.02887397	0.47920000	0.59020000
MOD2DEL	10	0.47410000	0.03815186	0.42440000	0.52890000
MRATE=7.5					
MOD3DEL	10	0.53511000	0.08405890	0.45400000	0.67840000

MOD2DEL	10	0.51384000	0.08334963	0.42010000	0.70420000
MRATE=8					
MOD3DEL	10	0.50192000	0.06237773	0.39830000	0.58480000
MOD2DEL	10	0.50483000	0.06060508	0.42530000	0.61630000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST A

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=9					
MOD3DEL	10	0.55183000	0.08021438	0.45110000	0.72240000
MOD2DEL	10	0.51793000	0.05759850	0.45890000	0.62000000
MRATE=10					
MOD3DEL	10	0.52945000	0.08121932	0.42330000	0.65850000
MOD2DEL	10	0.59332000	0.07664317	0.46420000	0.69890000
MRATE=11					
MOD3DEL	10	0.62684000	0.17572590	0.47510000	1.08400000
MOD2DEL	10	0.51793000	0.05759850	0.45890000	0.62000000
MRATE=12					
MOD3DEL	10	0.69207000	0.18496091	0.44520000	0.98750000
MOD2DEL	10	0.58507000	0.08735478	0.48990000	0.78870000
MRATE=13					
MOD3DEL	10	0.67448000	0.19964505	0.44570000	1.12700000
MOD2DEL	10	0.67036000	0.15556636	0.51370000	1.04700000
MRATE=14					
MOD3DEL	10	0.62193000	0.09991562	0.48280000	0.74850000
MOD2DEL	10	0.77107000	0.22005152	0.50040000	1.18150000
MRATE=15					
MOD3DEL	10	0.68206000	0.10390572	0.54870000	0.82980000
MOD2DEL	10	0.83504000	0.24317850	0.58280000	1.42600000
MRATE=16					
MOD3DEL	10	0.64249000	0.10097111	0.44080000	0.78020000
MOD2DEL	10	0.95874000	0.16737724	0.74420000	1.12650000
MRATE=17					

MOD3DEL	10	0.70610000	0.16504998	0.53260000	1.07700000
MOD2DEL	10	1.02129000	0.58040122	0.68770000	2.61100000

---------------------------- MRATE=18 ----------------------------

MOD3DEL	10	0.75124000	0.21768273	0.51970000	1.29800000
MOD2DEL	10	0.96365000	0.38730551	0.60170000	1.87600000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST A

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
---------- MRATE=19 ----------					
MOD3DEL	10	0.70303000	0.15703216	0.53120000	1.05100000
MOD2DEL	10	1.38597000	0.49365887	0.79970000	2.60700000
---------- MRATE=20 ----------					
MOD3DEL	10	0.87241000	0.33350955	0.37750000	1.52300000
MOD2DEL	10	1.77489000	0.73884770	0.88290000	3.59100000

Appendix: ANALYSIS OF SLAM DATA
Host B

MRATE=0.5

OBS	MOD2DEL	MOD3DEL	DIFF
1	0.4082	0.4021	-0.0061
2	0.3534	0.3245	-0.0289
3	0.3193	0.3771	0.0578
4	0.4430	0.3039	-0.1391
5	0.3918	0.3314	-0.0604
6	0.4064	0.4281	0.0217
7	0.3402	0.4074	0.0672
8	0.3195	0.4198	0.1003
9	0.3854	0.3074	-0.0780
10	0.3579	0.4224	0.0645

MRATE=1

OBS	MOD2DEL	MOD3DEL	DIFF
11	0.3769	0.4117	0.0348
12	0.4563	0.4686	0.0123
13	0.3773	0.3047	-0.0726
14	0.4094	0.3416	-0.0678
15	0.3739	0.4016	0.0277
16	0.3979	0.3017	-0.0962
17	0.4605	0.3752	-0.0853
18	0.3733	0.3204	-0.0529
19	0.4046	0.3519	-0.0527
20	0.3712	0.4415	0.0703

MRATE=2

OBS	MOD2DEL	MOD3DEL	DIFF
21	0.3823	0.4085	0.0262
22	0.4520	0.4045	-0.0475
23	0.3836	0.3445	-0.0391
24	0.4688	0.4809	0.0121
25	0.3949	0.3780	-0.0169
26	0.3784	0.3623	-0.0161
27	0.4358	0.3833	-0.0525
28	0.4159	0.3976	-0.0183
29	0.4113	0.3748	-0.0365
30	0.4819	0.3449	-0.1370

Appendix: ANALYSIS OF SLAM DATA
Host B

MRATE=3

OBS	MOD2DEL	MOD3DEL	DIFF
31	0.4397	0.4624	0.0227
32	0.4153	0.3875	-0.0278
33	0.3825	0.3899	0.0074
34	0.3742	0.4167	0.0425
35	0.4103	0.3911	-0.0192
36	0.3718	0.4714	0.0996
37	0.3733	0.3867	0.0134
38	0.3751	0.4248	0.0497
39	0.4011	0.4014	0.0003
40	0.4189	0.3887	-0.0302

MRATE=4

OBS	MOD2DEL	MOD3DEL	DIFF
41	0.4375	0.4352	-0.0023
42	0.4396	0.4349	-0.0047
43	0.4592	0.4307	-0.0285
44	0.4015	0.4247	0.0232
45	0.4150	0.3616	-0.0534
46	0.4373	0.4217	-0.0156
47	0.3954	0.3978	0.0024
48	0.4072	0.4194	0.0122
49	0.4285	0.4012	-0.0273
50	0.4355	0.3949	-0.0406

MRATE=5

OBS	MOD2DEL	MOD3DEL	DIFF
51	0.3972	0.4772	0.0800
52	0.4736	0.4115	-0.0621
53	0.4316	0.4751	0.0435
54	0.4064	0.4634	0.0570
55	0.3986	0.4028	0.0042
56	0.4364	0.6144	0.1780
57	0.4106	0.3901	-0.0205
58	0.4092	0.4056	-0.0036
59	0.4083	0.3947	-0.0136
60	0.4233	0.4392	0.0159

Appendix: ANALYSIS OF SLAM DATA
Host B

MRATE=6

OBS	MOD2DEL	MOD3DEL	DIFF
61	0.3726	0.5225	0.1499
62	0.3879	0.4422	0.0543
63	0.4070	0.4438	0.0368
64	0.3516	0.5858	0.2342
65	0.4160	0.4985	0.0825
66	0.3773	0.3930	0.0157
67	0.3512	0.5016	0.1504
68	0.4972	0.3910	-0.1062
69	0.3912	0.4778	0.0866
70	0.4478	0.4933	0.0455

MRATE=7

OBS	MOD2DEL	MOD3DEL	DIFF
71	0.4357	0.5588	0.1231
72	0.4626	0.4524	-0.0102
73	0.4710	0.4515	-0.0195
74	0.5410	0.6254	0.0844
75	0.4930	0.4695	-0.0235
76	0.5414	0.4682	-0.0732
77	0.4455	0.4612	0.0157
78	0.5134	0.4823	-0.0311
79	0.4694	0.4678	-0.0016
80	0.4743	0.5622	0.0879

MRATE=7.5

OBS	MOD2DEL	MOD3DEL	DIFF
81	0.6360	0.5705	-0.0655
82	0.4754	0.6299	0.1545
83	0.4494	0.5390	0.0896
84	0.5125	0.5734	0.0609
85	0.4201	0.5193	0.0992
86	0.5024	0.5621	0.0597
87	0.5277	0.5094	-0.0183
88	0.5588	0.5689	0.0101
89	0.4665	0.4300	-0.0365
90	0.4678	0.4763	0.0085

MRATE=8

OBS	MOD2DEL	MOD3DEL	DIFF
91	0.5225	0.5213	-0.0012
92	0.5257	0.5429	0.0172
93	0.4833	0.5585	0.0752
94	0.5295	0.7367	0.2072
95	0.4505	0.5144	0.0639
96	0.5780	0.4944	-0.0836
97	0.4879	0.4793	-0.0086
98	0.4531	0.5699	0.1168
99	0.5020	0.4706	-0.0314
100	0.4401	0.4836	0.0435

MRATE=9

OBS	MOD2DEL	MOD3DEL	DIFF
101	0.4854	0.5989	0.1135
102	0.4958	0.7314	0.2356
103	0.5338	0.5550	0.0212
104	0.5861	0.5404	-0.0457
105	0.5053	0.5761	0.0708
106	0.5220	0.5403	0.0183
107	0.5381	0.5739	0.0358
108	0.6039	0.5518	-0.0521
109	0.5101	0.6460	0.1359
110	0.4655	0.4737	0.0082

MRATE=10

OBS	MOD2DEL	MOD3DEL	DIFF
111	0.5428	0.6223	0.0795
112	0.5427	0.5016	-0.0411
113	0.5518	0.7689	0.2171
114	0.7429	0.7287	-0.0142
115	0.5352	1.0360	0.5008
116	0.7292	0.6259	-0.1033
117	0.6586	0.5200	-0.1386
118	0.6316	0.6399	0.0083
119	0.5567	0.9847	0.4280
120	0.4880	0.5351	0.0471
121	0.4854	0.7095	0.2241

Appendix: ANALYSIS OF SLAM DATA
Host B

MRATE=11

OBS	MOD2DEL	MOD3DEL	DIFF
122	0.4958	1.1170	0.6212
123	0.5338	0.6391	0.1053
124	0.5861	1.1080	0.5219
125	0.5053	0.5035	-0.0018
126	0.5220	0.6867	0.1647
127	0.5381	0.7837	0.2456
128	0.6039	0.7810	0.1771
129	0.5101	0.7283	0.2182
130	0.4655	0.7251	0.2596

MRATE=12

OBS	MOD2DEL	MOD3DEL	DIFF
131	0.6187	0.4995	-0.1192
132	0.5945	1.5080	0.9135
133	0.5458	0.8796	0.3338
134	0.5830	0.6930	0.1100
135	0.6013	0.8362	0.2349
136	0.5722	0.7542	0.1820
137	0.8201	2.1610	1.3409
138	0.5111	1.2890	0.7779
139	0.6605	1.1650	0.5045
140	0.6736	0.9389	0.2653

MRATE=13

OBS	MOD2DEL	MOD3DEL	DIFF
141	0.6185	0.8910	0.2725
142	0.7527	2.6950	1.9423
143	0.7374	0.6830	-0.0544
144	0.7020	1.7490	1.0470
145	0.5613	1.7790	1.2177
146	0.6948	0.8819	0.1871
147	0.8374	1.5380	0.7006
148	1.1510	0.7249	-0.4261
149	0.4982	0.8661	0.3679
150	0.8490	1.3200	0.4710

Appendix: ANALYSIS OF SLAM DATA
Host B

MRATE=14

OBS	MOD2DEL	MOD3DEL	DIFF
151	0.7866	0.9811	0.1945
152	1.2654	1.5250	0.2596
153	0.7537	1.1260	0.3723
154	0.6810	0.9938	0.3128
155	1.1181	1.9100	0.7919
156	0.5650	2.5040	1.9390
157	0.7694	2.1360	1.3666
158	0.5534	1.0280	0.4746
159	0.6086	0.9106	0.3020
160	0.7644	1.3010	0.5366

MRATE=15

OBS	MOD2DEL	MOD3DEL	DIFF
161	0.9117	2.746	1.8343
162	1.6380	2.212	0.5740
163	1.0390	2.750	1.7110
164	0.6678	1.227	0.5592
165	0.9289	4.681	3.7521
166	0.7544	1.303	0.5486
167	1.2450	1.470	0.2250
168	0.7304	1.385	0.6546
169	0.8401	3.421	2.5809
170	0.7027	3.048	2.3453

MRATE=16

OBS	MOD2DEL	MOD3DEL	DIFF
171	0.9447	3.333	2.3883
172	1.1064	3.590	2.4836
173	1.1328	2.255	1.1222
174	0.9111	1.348	0.4369
175	1.1004	1.994	0.8936
176	1.1536	2.026	0.8724
177	0.9438	4.333	3.3892
178	0.9712	1.276	0.3048
179	1.1037	2.115	1.0113
180	0.9853	1.851	0.8657

Appendix: ANALYSIS OF SLAM DATA
Host B

MRATE=17

OBS	MOD2DEL	MOD3DEL	DIFF
181	0.8658	3.346	2.4802
182	0.7125	2.653	1.9405
183	3.0120	4.035	1.0230
184	0.8587	2.285	1.4263
185	1.4400	1.651	0.2110
186	0.9927	3.535	2.5423
187	1.1870	3.411	2.2240
188	0.8238	1.868	1.0442
189	1.2140	4.038	2.8240
190	0.8721	2.147	1.2749

MRATE=18

OBS	MOD2DEL	MOD3DEL	DIFF
191	1.3870	3.165	1.7780
192	0.7106	4.362	3.6514
193	1.0340	4.374	3.3400
194	0.7263	2.808	2.0817
195	0.7449	3.201	2.4561
196	1.2820	5.130	3.8480
197	1.0880	2.093	1.0050
198	0.8830	5.119	4.2360
199	0.9791	4.292	3.3129
200	2.1590	4.157	1.9980

MRATE=19

OBS	MOD2DEL	MOD3DEL	DIFF
201	1.6910	3.856	2.1650
202	1.9710	3.735	1.7640
203	0.9003	3.940	3.0397
204	1.3440	3.623	2.2790
205	1.3560	3.685	2.3290
206	3.5270	4.046	0.5190
207	1.6600	4.750	3.0900
208	1.4370	2.830	1.3930
209	1.8590	4.369	2.5100
210	1.6920	4.071	2.3790

Appendix: ANALYSIS OF SLAM DATA
Host B

------------------------------ MRATE=20 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
211	1.245	6.842	5.597
212	1.380	4.399	3.019
213	3.249	5.431	2.182
214	2.261	3.058	0.797
215	1.456	4.700	3.244
216	2.341	5.733	3.392
217	2.482	6.919	4.437
218	2.224	7.405	5.181
219	1.947	6.110	4.163
220	1.975	1.155	-0.820

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST B

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>¦T¦
MRATE=0.5				
DIFF	-0.00010000	0.02415257	-0.00	0.9968
MRATE=1				
DIFF	-0.02824000	0.01858033	-1.52	0.1629
MRATE=2				
DIFF	-0.03256000	0.01404500	-2.32	0.0456
MRATE=3				
DIFF	0.01584000	0.01268887	1.25	0.2434
MRATE=4				
DIFF	-0.01346000	0.00760798	-1.77	0.1106
MRATE=5				
DIFF	0.02788000	0.02113379	1.32	0.2197
MRATE=6				
DIFF	0.07497000	0.02904757	2.58	0.0297
MRATE=7				
DIFF	0.01520000	0.01980573	0.77	0.4625
MRATE=7.5				
DIFF	0.03622000	0.02160882	1.68	0.1280
MRATE=8				
DIFF	0.03990000	0.02595523	1.54	0.1586
MRATE=9				
DIFF	0.05415000	0.02778588	1.95	0.0831

MRATE=10				
DIFF	0.10979091	0.06318247	1.74	0.1129
MRATE=11				
DIFF	0.25686667	0.06554783	3.92	0.0044

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST B

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>|T|
MRATE=12				
DIFF	0.45436000	0.13840885	3.28	0.0095
MRATE=13				
DIFF	0.57256000	0.21641260	2.65	0.0267
MRATE=14				
DIFF	0.65499000	0.17982601	3.64	0.0054
MRATE=15				
DIFF	1.47850000	0.36658416	4.03	0.0030
MRATE=16				
DIFF	1.37680000	0.32107501	4.29	0.0020
MRATE=17				
DIFF	1.69904000	0.26405197	6.43	0.0001
MRATE=18				
DIFF	2.77071000	0.33292149	8.32	0.0001
MRATE=19				
DIFF	2.14677000	0.24240040	8.86	0.0001
MRATE=20				
DIFF	3.11920000	0.62454353	4.99	0.0007

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST B

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
---------- MRATE=0.5 ----------					
MOD3DEL	10	0.37241000	0.05041063	0.30390000	0.42810000
MOD2DEL	10	0.37251000	0.04108710	0.31930000	0.44300000
---------- MRATE=1 ----------					
MOD3DEL	10	0.37189000	0.05781800	0.30170000	0.46860000
MOD2DEL	10	0.40013000	0.03369164	0.37120000	0.46050000
---------- MRATE=2 ----------					
MOD3DEL	10	0.38793000	0.03961159	0.34450000	0.48090000
MOD2DEL	10	0.42049000	0.03754051	0.37840000	0.48190000
---------- MRATE=3 ----------					
MOD3DEL	10	0.41206000	0.03175466	0.38670000	0.47140000
MOD2DEL	10	0.39622000	0.02411048	0.37180000	0.43970000
---------- MRATE=4 ----------					
MOD3DEL	10	0.41221000	0.02326869	0.36160000	0.43520000
MOD2DEL	10	0.42567000	0.02014834	0.39540000	0.45920000
---------- MRATE=5 ----------					
MOD3DEL	10	0.44740000	0.06745646	0.39010000	0.61440000
MOD2DEL	10	0.41952000	0.02306926	0.39720000	0.47360000
---------- MRATE=6 ----------					
MOD3DEL	10	0.47495000	0.05959557	0.39100000	0.58580000
MOD2DEL	10	0.39998000	0.04503280	0.35120000	0.49720000
---------- MRATE=7 ----------					
MOD3DEL	10	0.49993000	0.06006151	0.45150000	0.62540000
MOD2DEL	10	0.48473000	0.03686784	0.43570000	0.54140000
---------- MRATE=7.5 ----------					
MOD3DEL	10	0.53788000	0.05669611	0.43000000	0.62990000

MOD2DEL	10	0.50166000	0.06197114	0.42010000	0.63600000

------------------------------- MRATE=8 -------------------------------

MOD3DEL	10	0.53716000	0.07790388	0.47060000	0.73670000
MOD2DEL	10	0.49726000	0.04309303	0.44010000	0.57800000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST B

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
---------- MRATE=9 ----------					
MOD3DEL	10	0.57875000	0.06955106	0.47370000	0.73140000
MOD2DEL	10	0.52460000	0.04318696	0.46550000	0.60390000
---------- MRATE=10 ----------					
MOD3DEL	11	0.69750909	0.17709818	0.50160000	1.03600000
MOD2DEL	11	0.58771818	0.08973057	0.48540000	0.74290000
---------- MRATE=11 ----------					
MOD3DEL	9	0.78582222	0.20358028	0.50350000	1.11700000
MOD2DEL	9	0.52895556	0.04341463	0.46550000	0.60390000
---------- MRATE=12 ----------					
MOD3DEL	10	1.07244000	0.48470864	0.49950000	2.16100000
MOD2DEL	10	0.61808000	0.08598235	0.51110000	0.82010000
---------- MRATE=13 ----------					
MOD3DEL	10	1.31279000	0.63894548	0.68300000	2.69500000
MOD2DEL	10	0.74023000	0.18230210	0.49820000	1.15100000
---------- MRATE=14 ----------					
MOD3DEL	10	1.44155000	0.55998234	0.91060000	2.50400000
MOD2DEL	10	0.78656000	0.23269840	0.55340000	1.26540000
---------- MRATE=15 ----------					
MOD3DEL	10	2.42430000	1.12602665	1.22700000	4.68100000
MOD2DEL	10	0.94580000	0.29988542	0.66780000	1.63800000
---------- MRATE=16 ----------					
MOD3DEL	10	2.41210000	1.00554335	1.27600000	4.33300000
MOD2DEL	10	1.03530000	0.09194683	0.91110000	1.15360000
---------- MRATE=17 ----------					
MOD3DEL	10	2.89690000	0.88679842	1.65100000	4.03800000

MOD2DEL	10	1.19786000	0.67487554	0.71250000	3.01200000

------------------------------ MRATE=18 ------------------------------

MOD3DEL	10	3.87010000	1.00743309	2.09300000	5.13000000
MOD2DEL	10	1.09939000	0.43723324	0.71060000	2.15900000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST B

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=19					
MOD3DEL	10	3.89050000	0.50450531	2.83000000	4.75000000
MOD2DEL	10	1.74373000	0.69706001	0.90030000	3.52700000
MRATE=20					
MOD3DEL	10	5.17520000	1.92971902	1.15500000	7.40500000
MOD2DEL	10	2.05600000	0.60086401	1.24500000	3.24900000

Appendix: ANALYSIS OF SLAM DATA
HOST C

------------------------------ MRATE=0.5 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
1	0.3892	0.4311	0.0419
2	0.3795	0.4687	0.0892
3	0.3901	0.4898	0.0997
4	0.4328	0.4650	0.0322
5	0.4275	0.4744	0.0469
6	0.4061	0.4252	0.0191
7	0.4222	0.4353	0.0131
8	0.3883	0.4238	0.0355
9	0.4047	0.4058	0.0011
10	0.4156	0.4846	0.0690

------------------------------ MRATE=1 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
11	0.4266	0.4945	0.0679
12	0.4086	0.4410	0.0324
13	0.4035	0.5110	0.1075
14	0.4638	0.5037	0.0399
15	0.4440	0.4733	0.0293
16	0.4871	0.4036	-0.0835
17	0.4036	0.4703	0.0667
18	0.3889	0.3728	-0.0161
19	0.4686	0.3985	-0.0701
20	0.3948	0.4731	0.0783

------------------------------ MRATE=2 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
21	0.4033	0.4650	0.0617
22	0.4402	0.5106	0.0704
23	0.4376	0.4547	0.0171
24	0.4234	0.5003	0.0769
25	0.4215	0.4457	0.0242
26	0.4329	0.4602	0.0273
27	0.4826	0.4555	-0.0271
28	0.4507	0.4287	-0.0220
29	0.4115	0.4729	0.0614
30	0.4170	0.5029	0.0859

Appendix: ANALYSIS OF SLAM DATA
HOST C

MRATE=3

OBS	MOD2DEL	MOD3DEL	DIFF
31	0.4168	0.4611	0.0443
32	0.4564	0.4189	-0.0375
33	0.4196	0.5799	0.1603
34	0.4071	0.5362	0.1291
35	0.4042	0.4726	0.0684
36	0.3974	0.4877	0.0903
37	0.4514	0.4565	0.0051
38	0.4205	0.4314	0.0109
39	0.4336	0.4733	0.0397
40	0.3818	0.5183	0.1365

MRATE=4

OBS	MOD2DEL	MOD3DEL	DIFF
41	0.4254	0.4067	-0.0187
42	0.4485	0.4786	0.0301
43	0.4281	0.4031	-0.0250
44	0.4107	0.5566	0.1459
45	0.4151	0.5137	0.0986
46	0.4845	0.4676	-0.0169
47	0.4245	0.4699	0.0454
48	0.4160	0.4671	0.0511
49	0.4414	0.4385	-0.0029
50	0.4543	0.4621	0.0078

MRATE=5

OBS	MOD2DEL	MOD3DEL	DIFF
51	0.4694	0.4256	-0.0438
52	0.4541	0.5064	0.0523
53	0.4298	0.5725	0.1427
54	0.4670	0.5164	0.0494
55	0.4643	0.4871	0.0228
56	0.4673	0.4500	-0.0173
57	0.4029	0.4171	0.0142
58	0.4567	0.4217	-0.0350
59	0.4180	0.4576	0.0396
60	0.4306	0.5162	0.0856

Appendix: ANALYSIS OF SLAM DATA
HOST C

MRATE=6

OBS	MOD2DEL	MOD3DEL	DIFF
61	0.4047	0.5527	0.1480
62	0.4316	0.5798	0.1482
63	0.4080	0.6180	0.2100
64	0.4417	0.5363	0.0946
65	0.4208	0.5588	0.1380
66	0.3898	0.5379	0.1481
67	0.4171	0.4602	0.0431
68	0.4244	0.5016	0.0772
69	0.4778	0.6212	0.1434
70	0.4376	0.4808	0.0432

MRATE=7

OBS	MOD2DEL	MOD3DEL	DIFF
71	0.4417	0.5803	0.1386
72	0.4069	0.4543	0.0474
73	0.5048	0.5900	0.0852
74	0.5403	0.6011	0.0608
75	0.4830	0.4280	-0.0550
76	0.5364	0.6533	0.1169
77	0.4738	0.4855	0.0117
78	0.4734	0.5405	0.0671
79	0.5051	0.4825	-0.0226
80	0.4309	0.4971	0.0662

MRATE=7.5

OBS	MOD2DEL	MOD3DEL	DIFF
81	0.4856	0.6372	0.1516
82	0.4784	0.5531	0.0747
83	0.4606	0.4966	0.0360
84	0.5183	0.5697	0.0514
85	0.4501	0.4350	-0.0151
86	0.4718	0.5246	0.0528
87	0.5169	0.4795	-0.0374
88	0.6341	0.5455	-0.0886
89	0.4904	0.5755	0.0851
90	0.5191	0.5272	0.0081

Appendix: ANALYSIS OF SLAM DATA
HOST C

MRATE=8

OBS	MOD2DEL	MOD3DEL	DIFF
91	0.4780	0.5275	0.0495
92	0.5932	0.4498	-0.1434
93	0.4864	0.4894	0.0030
94	0.5701	0.4982	-0.0719
95	0.4581	0.4512	-0.0069
96	0.6185	0.4634	-0.1551
97	0.5082	0.6135	0.1053
98	0.4821	0.4628	-0.0193
99	0.4670	0.4799	0.0129
100	0.3998	0.5378	0.1380

MRATE=9

OBS	MOD2DEL	MOD3DEL	DIFF
101	0.4338	0.4635	0.0297
102	0.5468	0.6423	0.0955
103	0.5239	0.4685	-0.0554
104	0.5384	0.5489	0.0105
105	0.5799	0.5802	0.0003
106	0.4732	0.4892	0.0160
107	0.5642	0.6277	0.0635
108	0.5360	0.5427	0.0067
109	0.5376	0.4986	-0.0390
110	0.4475	0.5035	0.0560

MRATE=10

OBS	MOD2DEL	MOD3DEL	DIFF
111	0.4951	0.5755	0.0804
112	0.5440	0.5820	0.0380
113	0.5041	0.7235	0.2194
114	0.5804	0.4451	-0.1353
115	0.4811	0.5072	0.0261
116	0.5724	0.5172	-0.0552
117	0.6064	0.6079	0.0015
118	0.5511	0.5423	-0.0088
119	0.5459	0.5380	-0.0079
120	0.4800	0.5248	0.0448

Appendix: ANALYSIS OF SLAM DATA
HOST C

MRATE=11

OBS	MOD2DEL	MOD3DEL	DIFF
121	0.4338	0.5814	0.1476
122	0.5468	1.1840	0.6372
123	0.5239	0.5188	-0.0051
124	0.5384	0.4714	-0.0670
125	0.5799	0.4602	-0.1197
126	0.4732	0.4345	-0.0387
127	0.5642	0.8688	0.3046
128	0.5360	0.4865	-0.0495
129	0.5376	0.7227	0.1851
130	0.4475	0.5098	0.0623

MRATE=12

OBS	MOD2DEL	MOD3DEL	DIFF
131	0.5534	0.5930	0.0396
132	0.4730	0.7082	0.2352
133	0.5162	0.6364	0.1202
134	0.5114	0.4543	-0.0571
135	0.5583	0.7482	0.1899
136	0.4822	0.5896	0.1074
137	0.7944	0.6293	-0.1651
138	0.4544	1.0360	0.5816
139	0.6113	0.6337	0.0224
140	0.5716	0.5094	-0.0622

MRATE=13

OBS	MOD2DEL	MOD3DEL	DIFF
141	0.6168	0.5615	-0.0553
142	0.6275	0.7266	0.0991
143	0.5767	0.5573	-0.0194
144	0.6904	0.5796	-0.1108
145	0.4992	0.7218	0.2226
146	0.6235	0.5195	-0.1040
147	0.6707	0.6717	0.0010
148	0.9022	0.5386	-0.3636
149	0.5119	0.5143	0.0024
150	0.7053	0.7312	0.0259

Appendix: ANALYSIS OF SLAM DATA
HOST C

MRATE=14

OBS	MOD2DEL	MOD3DEL	DIFF
151	0.6607	0.5295	-0.1312
152	1.1310	0.7609	-0.3701
153	0.4677	0.5112	0.0435
154	0.6647	0.5958	-0.0689
155	0.6622	0.6659	0.0037
156	0.5968	0.7248	0.1280
157	0.5889	0.8321	0.2432
158	0.5954	0.4699	-0.1255
159	0.5699	0.6122	0.0423
160	0.7512	0.6731	-0.0781

MRATE=15

OBS	MOD2DEL	MOD3DEL	DIFF
161	0.7144	0.7105	-0.0039
162	0.8104	0.6206	-0.1898
163	0.7066	0.7077	0.0011
164	0.6115	0.5548	-0.0567
165	0.6937	0.7610	0.0673
166	0.6578	0.5565	-0.1013
167	0.8465	0.8062	-0.0403
168	0.6070	0.7170	0.1100
169	0.6464	0.6795	0.0331
170	0.6685	0.9005	0.2320

MRATE=16

OBS	MOD2DEL	MOD3DEL	DIFF
171	0.6458	0.7171	0.0713
172	0.8416	0.6694	-0.1722
173	0.9316	0.7443	-0.1873
174	0.7500	0.5816	-0.1684
175	0.6795	0.5227	-0.1568
176	1.1106	0.8148	-0.2958
177	0.8136	0.8955	0.0819
178	0.8213	0.5946	-0.2267
179	0.6468	0.8177	0.1709
180	0.9979	0.7286	-0.2693

Appendix: ANALYSIS OF SLAM DATA
HOST C

MRATE=17

OBS	MOD2DEL	MOD3DEL	DIFF
181	0.7597	0.7737	0.0140
182	0.6809	0.7974	0.1165
183	1.8310	1.3990	-0.4320
184	0.5701	0.6233	0.0532
185	0.7693	0.4973	-0.2720
186	0.7954	0.8381	0.0427
187	0.8836	0.6539	-0.2297
188	0.6260	0.4014	-0.2246
189	0.9827	0.7156	-0.2671
190	0.6341	0.6189	-0.0152

MRATE=18

OBS	MOD2DEL	MOD3DEL	DIFF
191	0.9337	0.7485	-0.1852
192	0.6795	0.5991	-0.0804
193	0.7815	0.7572	-0.0243
194	0.5484	0.6911	0.1427
195	0.6404	0.7295	0.0891
196	0.9913	1.1430	0.1517
197	1.0110	0.6700	-0.3410
198	0.7101	1.6510	0.9409
199	0.7669	0.7296	-0.0373
200	1.5310	1.0090	-0.5220

MRATE=19

OBS	MOD2DEL	MOD3DEL	DIFF
201	1.1750	0.7578	-0.4172
202	1.2320	0.7341	-0.4979
203	0.8106	0.6590	-0.1516
204	0.8242	0.8974	0.0732
205	1.3080	0.8336	-0.4744
206	1.9960	1.0250	-0.9710
207	1.0230	0.8948	-0.1282
208	1.1190	0.6234	-0.4956
209	0.8859	1.0240	0.1381
210	1.1090	1.1680	0.0590

Appendix: ANALYSIS OF SLAM DATA
HOST C

------------------------------ MRATE=20 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
211	1.2170	0.9361	-0.2809
212	0.9788	0.6021	-0.3767
213	1.9900	0.6744	-1.3156
214	1.2770	0.6864	-0.5906
215	0.8755	0.6214	-0.2541
216	1.4120	1.5270	0.1150
217	1.4030	1.2520	-0.1510
218	1.3310	0.8736	-0.4574
219	1.2040	0.6365	-0.5675
220	1.3370	0.7485	-0.5885

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST C

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>¦T¦
MRATE=0.5				
DIFF	0.04477000	0.01022206	4.38	0.0018
MRATE=1				
DIFF	0.02523000	0.02000740	1.26	0.2390
MRATE=2				
DIFF	0.03758000	0.01271247	2.96	0.0161
MRATE=3				
DIFF	0.06471000	0.02028782	3.19	0.0110
MRATE=4				
DIFF	0.03154000	0.01762857	1.79	0.1072
MRATE=5				
DIFF	0.03105000	0.01793744	1.73	0.1175
MRATE=6				
DIFF	0.11938000	0.01687385	7.07	0.0001
MRATE=7				
DIFF	0.05163000	0.01884998	2.74	0.0229
MRATE=7.5				
DIFF	0.03186000	0.02158245	1.48	0.1740
MRATE=8				
DIFF	-0.00879000	0.03022037	-0.29	0.7777
MRATE=9				
DIFF	0.01838000	0.01445058	1.27	0.2353

---------------------- MRATE=10 ----------------------

DIFF	0.02030000	0.02907979	0.70	0.5028

---------------------- MRATE=11 ----------------------

DIFF	0.10568000	0.07225283	1.46	0.1776

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST C

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>¦T¦
MRATE=12				
DIFF	0.10119000	0.06580151	1.54	0.1585
MRATE=13				
DIFF	-0.03021000	0.04826487	-0.63	0.5469
MRATE=14				
DIFF	-0.03131000	0.05272642	-0.59	0.5673
MRATE=15				
DIFF	0.00515000	0.03688191	0.14	0.8920
MRATE=16				
DIFF	-0.11524000	0.05134713	-2.24	0.0515
MRATE=17				
DIFF	-0.12142000	0.05834593	-2.08	0.0672
MRATE=18				
DIFF	0.01342000	0.12321241	0.11	0.9157
MRATE=19				
DIFF	-0.28656000	0.10962757	-2.61	0.0281
MRATE=20				
DIFF	-0.44673000	0.11939377	-3.74	0.0046

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST C

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=0.5					
MOD3DEL	10	0.45037000	0.02939823	0.40580000	0.48980000
MOD2DEL	10	0.40560000	0.01851660	0.37950000	0.43280000
MRATE=1					
MOD3DEL	10	0.45418000	0.04801025	0.37280000	0.51100000
MOD2DEL	10	0.42895000	0.03476071	0.38890000	0.48710000
MRATE=2					
MOD3DEL	10	0.46965000	0.02691056	0.42870000	0.51060000
MOD2DEL	10	0.43207000	0.02276098	0.40330000	0.48260000
MRATE=3					
MOD3DEL	10	0.48359000	0.04903517	0.41890000	0.57990000
MOD2DEL	10	0.41888000	0.02328575	0.38180000	0.45640000
MRATE=4					
MOD3DEL	10	0.46639000	0.04583088	0.40310000	0.55660000
MOD2DEL	10	0.43485000	0.02271379	0.41070000	0.48450000
MRATE=5					
MOD3DEL	10	0.47706000	0.05117400	0.41710000	0.57250000
MOD2DEL	10	0.44601000	0.02380114	0.40290000	0.46940000
MRATE=6					
MOD3DEL	10	0.54473000	0.05366338	0.46020000	0.62120000
MOD2DEL	10	0.42535000	0.02424038	0.38980000	0.47780000
MRATE=7					
MOD3DEL	10	0.53126000	0.07298094	0.42800000	0.65330000
MOD2DEL	10	0.47963000	0.04393561	0.40690000	0.54030000
MRATE=7.5					
MOD3DEL	10	0.53439000	0.05622092	0.43500000	0.63720000

MOD2DEL	10	0.50253000	0.05218323	0.45010000	0.63410000

------------------------------ MRATE=8 ------------------------------

MOD3DEL	10	0.49735000	0.05070788	0.44980000	0.61350000
MOD2DEL	10	0.50614000	0.06766483	0.39980000	0.61850000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST C

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=9					
MOD3DEL	10	0.53651000	0.06353360	0.46350000	0.64230000
MOD2DEL	10	0.51813000	0.04946534	0.43380000	0.57990000
MRATE=10					
MOD3DEL	10	0.55635000	0.07414378	0.44510000	0.72350000
MOD2DEL	10	0.53605000	0.04407431	0.48000000	0.60640000
MRATE=11					
MOD3DEL	10	0.62381000	0.23900849	0.43450000	1.18400000
MOD2DEL	10	0.51813000	0.04946534	0.43380000	0.57990000
MRATE=12					
MOD3DEL	10	0.65381000	0.15906828	0.45430000	1.03600000
MOD2DEL	10	0.55262000	0.09790302	0.45440000	0.79440000
MRATE=13					
MOD3DEL	10	0.61221000	0.09008804	0.51430000	0.73120000
MOD2DEL	10	0.64242000	0.11433134	0.49920000	0.90220000
MRATE=14					
MOD3DEL	10	0.63754000	0.11589092	0.46990000	0.83210000
MOD2DEL	10	0.66885000	0.17879925	0.46770000	1.13100000
MRATE=15					
MOD3DEL	10	0.70143000	0.10751624	0.55480000	0.90050000
MOD2DEL	10	0.69628000	0.07883170	0.60700000	0.84650000
MRATE=16					
MOD3DEL	10	0.70863000	0.11788886	0.52270000	0.89550000
MOD2DEL	10	0.82387000	0.15416258	0.64580000	1.11060000
MRATE=17					
MOD3DEL	10	0.73186000	0.27029169	0.40140000	1.39900000

MOD2DEL	10	0.85328000	0.36550833	0.57010000	1.83100000

------------------------------ MRATE=18 ------------------------------

MOD3DEL	10	0.87280000	0.31903729	0.59910000	1.65100000
MOD2DEL	10	0.85938000	0.28105128	0.54840000	1.53100000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST C

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=19					
MOD3DEL	10	0.86171000	0.17478200	0.62340000	1.16800000
MOD2DEL	10	1.14827000	0.34284057	0.81060000	1.99600000
MRATE=20					
MOD3DEL	10	0.85580000	0.30817815	0.60210000	1.52700000
MOD2DEL	10	1.30253000	0.29842600	0.87550000	1.99000000

Appendix: ANALYSIS OF SLAM DATA.
Host D

------------------------------ MRATE=0.5 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
1	0.4095	0.4247	0.0152
2	0.3980	0.4088	0.0108
3	0.3916	0.4376	0.0460
4	0.4562	0.4133	-0.0429
5	0.3974	0.4730	0.0756
6	0.3781	0.4493	0.0712
7	0.4144	0.5180	0.1036
8	0.4165	0.3880	-0.0285
9	0.3952	0.4091	0.0139
10	0.4233	0.4846	0.0613

------------------------------ MRATE=1 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
11	0.3988	0.4347	0.0359
12	0.4254	0.4562	0.0308
13	0.4097	0.4679	0.0582
14	0.4693	0.4615	-0.0078
15	0.4272	0.4750	0.0478
16	0.4689	0.4090	-0.0599
17	0.4017	0.4770	0.0753
18	0.3732	0.4631	0.0899
19	0.4232	0.4417	0.0185
20	0.3913	0.4931	0.1018

------------------------------ MRATE=2 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
21	0.4097	0.4112	0.0015
22	0.4120	0.4396	0.0276
23	0.3901	0.4534	0.0633
24	0.4594	0.5112	0.0518
25	0.4363	0.3963	-0.0400
26	0.3945	0.4765	0.0820
27	0.4888	0.4505	-0.0383
28	0.4163	0.4754	0.0591
29	0.3933	0.4192	0.0259
30	0.3958	0.4266	0.0308

Appendix: ANALYSIS OF SLAM DATA.
Host D

MRATE=3

OBS	MOD2DEL	MOD3DEL	DIFF
31	0.4327	0.4282	-0.0045
32	0.4839	0.4147	-0.0692
33	0.4552	0.6160	0.1608
34	0.4221	0.4653	0.0432
35	0.3925	0.4210	0.0285
36	0.4261	0.4775	0.0514
37	0.4254	0.4299	0.0045
38	0.4448	0.4262	-0.0186
39	0.4563	0.5896	0.1333
40	0.4148	0.4129	-0.0019

MRATE=4

OBS	MOD2DEL	MOD3DEL	DIFF
41	0.4440	0.5863	0.1423
42	0.4410	0.5961	0.1551
43	0.4728	0.4074	-0.0654
44	0.3950	0.4975	0.1025
45	0.4095	0.4802	0.0707
46	0.4136	0.4532	0.0396
47	0.4218	0.4810	0.0592
48	0.4177	0.5210	0.1033
49	0.4510	0.5204	0.0694
50	0.4122	0.4973	0.0851

MRATE=5

OBS	MOD2DEL	MOD3DEL	DIFF
51	0.4277	0.5198	0.0921
52	0.4525	0.4729	0.0204
53	0.4505	0.5851	0.1346
54	0.4591	0.4994	0.0403
55	0.4525	0.4870	0.0345
56	0.4961	0.4856	-0.0105
57	0.3893	0.4594	0.0701
58	0.5514	0.4309	-0.1205
59	0.3989	0.4905	0.0916
60	0.4426	0.4315	-0.0111

Appendix: ANALYSIS OF SLAM DATA.
Host D

MRATE=6

OBS	MOD2DEL	MOD3DEL	DIFF
61	0.3987	0.4511	0.0524
62	0.4111	0.4542	0.0431
63	0.4005	0.4914	0.0909
64	0.4385	0.7284	0.2899
65	0.4381	0.4754	0.0373
66	0.4188	0.5134	0.0946
67	0.4123	0.5046	0.0923
68	0.4197	0.5005	0.0808
69	0.4413	0.5420	0.1007
70	0.4539	0.5695	0.1156

MRATE=7

OBS	MOD2DEL	MOD3DEL	DIFF
71	0.4219	0.6696	0.2477
72	0.4229	0.5318	0.1089
73	0.4820	0.5572	0.0752
74	0.5501	0.8420	0.2919
75	0.4501	0.4729	0.0228
76	0.5370	0.4987	-0.0383
77	0.4646	0.5849	0.1203
78	0.4604	0.5702	0.1098
79	0.5378	0.5560	0.0182
80	0.4354	0.4853	0.0499

MRATE=7.5

OBS	MOD2DEL	MOD3DEL	DIFF
81	0.4940	0.5398	0.0458
82	0.4763	0.5307	0.0544
83	0.4446	0.5143	0.0697
84	0.5417	0.5117	-0.0300
85	0.4210	0.5467	0.1257
86	0.4675	0.5544	0.0869
87	0.5646	0.4846	-0.0800
88	0.6567	0.5157	-0.1410
89	0.4204	0.4273	0.0069
90	0.5196	0.4884	-0.0312

Appendix: ANALYSIS OF SLAM DATA.
Host D

------------------------------ MRATE=8 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
91	0.5033	0.4720	-0.0313
92	0.5807	0.4633	-0.1174
93	0.4979	0.4649	-0.0330
94	0.5179	0.6161	0.0982
95	0.4409	0.5025	0.0616
96	0.5821	0.4395	-0.1426
97	0.5200	0.5595	0.0395
98	0.4830	0.4583	-0.0247
99	0.4599	0.4328	-0.0271
100	0.4131	0.6273	0.2142

------------------------------ MRATE=9 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
101	0.4769	0.4669	-0.0100
102	0.5166	0.4871	-0.0295
103	0.5358	0.5181	-0.0177
104	0.5482	0.4615	-0.0867
105	0.5993	0.4867	-0.1126
106	0.4966	0.4854	-0.0112
107	0.5729	0.5466	-0.0263
108	0.5709	0.6269	0.0560
109	0.5906	0.4821	-0.1085
110	0.4886	0.5231	0.0345

------------------------------ MRATE=10 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
111	0.5296	0.6047	0.0751
112	0.5622	0.5825	0.0203
113	0.5173	0.5845	0.0672
114	0.5886	0.5687	-0.0199
115	0.4994	0.7090	0.2096
116	0.5628	0.5064	-0.0564
117	0.6147	0.5145	-0.1002
118	0.6011	0.4534	-0.1477
119	0.4884	0.7075	0.2191
120	0.4552	0.5053	0.0501

Appendix: ANALYSIS OF SLAM DATA.
Host D

------------------------------ MRATE=11 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
121	0.4769	0.4976	0.0207
122	0.5166	0.6508	0.1342
123	0.5358	0.4308	-0.1050
124	0.5482	0.4620	-0.0862
125	0.5993	0.4375	-0.1618
126	0.4966	0.5189	0.0223
127	0.5729	0.5988	0.0259
128	0.5709	0.6965	0.1256
129	0.5906	0.5438	-0.0468
130	0.4886	0.5647	0.0761

------------------------------ MRATE=12 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
131	0.6533	0.7427	0.0894
132	0.5133	0.8541	0.3408
133	0.4985	1.8920	1.3935
134	0.4968	0.5376	0.0408
135	0.5794	0.5875	0.0081
136	0.5057	0.5020	-0.0037
137	0.8282	0.7568	-0.0714
138	0.4564	1.5560	1.0996
139	0.6052	0.7979	0.1927
140	0.5841	1.3020	0.7179

------------------------------ MRATE=13 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
141	0.5598	0.5840	0.0242
142	0.6695	0.7403	0.0708
143	0.6497	0.6660	0.0163
144	0.6028	0.7410	0.1382
145	0.4843	0.5683	0.0840
146	0.5717	0.4948	-0.0769
147	0.5823	0.6862	0.1039
148	1.0220	0.4775	-0.5445
149	0.5335	0.5898	0.0563
150	0.7381	0.5586	-0.1795

Appendix: ANALYSIS OF SLAM DATA.
Host D

MRATE=14

OBS	MOD2DEL	MOD3DEL	DIFF
151	0.7722	0.5421	-0.2301
152	1.1441	0.5811	-0.5630
153	0.5502	0.6244	0.0742
154	0.6830	0.7045	0.0215
155	0.7923	0.7177	-0.0746
156	0.5606	0.5206	-0.0400
157	0.6359	0.5664	-0.0695
158	0.5195	0.6327	0.1132
159	0.5286	0.6891	0.1605
160	0.7207	0.7177	-0.0030

MRATE=15

OBS	MOD2DEL	MOD3DEL	DIFF
161	0.6675	0.5352	-0.1323
162	0.8500	0.6447	-0.2053
163	0.8348	0.5610	-0.2738
164	0.5858	0.5204	-0.0654
165	0.8538	1.0240	0.1702
166	0.6130	0.7076	0.0946
167	1.0410	0.5593	-0.4817
168	0.6220	1.0120	0.3900
169	0.6811	0.6456	-0.0355
170	0.5833	0.8064	0.2231

MRATE=16

OBS	MOD2DEL	MOD3DEL	DIFF
171	0.7358	0.7260	-0.0098
172	0.7595	0.7549	-0.0046
173	0.8652	0.6251	-0.2401
174	0.7647	1.1270	0.3623
175	0.6506	0.5272	-0.1234
176	0.9114	1.3380	0.4266
177	0.7761	0.7884	0.0123
178	0.7125	1.0370	0.3245
179	0.7178	1.1370	0.4192
180	0.9325	0.8865	-0.0460

MRATE=17

OBS	MOD2DEL	MOD3DEL	DIFF
181	0.6255	0.6472	0.0217
182	0.6769	0.8239	0.1470
183	1.5510	1.0140	-0.5370
184	0.6105	0.6160	0.0055
185	0.7509	0.5226	-0.2283
186	0.6719	0.7600	0.0881
187	0.9679	0.8882	-0.0797
188	0.6652	0.5522	-0.1130
189	0.9229	0.6400	-0.2829
190	0.6976	0.4711	-0.2265

MRATE=18

OBS	MOD2DEL	MOD3DEL	DIFF
191	0.8933	0.7525	-0.1408
192	0.6617	0.5972	-0.0645
193	0.8333	0.6086	-0.2247
194	0.5857	0.5998	0.0141
195	0.6362	0.5721	-0.0641
196	0.9538	0.7806	-0.1732
197	1.0200	0.5899	-0.4301
198	0.6829	0.9500	0.2671
199	0.9261	1.0030	0.0769
200	1.4770	0.5228	-0.9542

MRATE=19

OBS	MOD2DEL	MOD3DEL	DIFF
201	1.0540	0.7060	-0.3480
202	0.9905	0.8132	-0.1773
203	0.8341	0.6290	-0.2051
204	0.8890	0.7406	-0.1484
205	1.0400	0.7122	-0.3278
206	2.4620	0.7365	-1.7255
207	1.1980	0.6219	-0.5761
208	1.0100	0.5391	-0.4709
209	1.2040	0.7537	-0.4503
210	0.9427	1.1130	0.1703

Appendix: ANALYSIS OF SLAM DATA.
Host D

------------------------------ MRATE=20 ------------------------------

OBS	MOD2DEL	MOD3DEL	DIFF
211	0.8972	1.2630	0.3658
212	0.8217	0.5262	-0.2955
213	2.0490	0.6694	-1.3796
214	1.3860	0.5955	-0.7905
215	1.1360	0.7007	-0.4353
216	1.2200	0.6036	-0.6164
217	2.0410	0.9582	-1.0828
218	1.2390	0.9436	-0.2954
219	0.9996	0.5920	-0.4076
220	1.0410	0.9519	-0.0891

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST D

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>¦T¦
MRATE=0.5				
DIFF	0.03262000	0.01488637	2.19	0.0561
MRATE=1				
DIFF	0.03905000	0.01519025	2.57	0.0302
MRATE=2				
DIFF	0.02637000	0.01308244	2.02	0.0746
MRATE=3				
DIFF	0.03275000	0.02195932	1.49	0.1701
MRATE=4				
DIFF	0.07618000	0.01939511	3.93	0.0035
MRATE=5				
DIFF	0.03415000	0.02266327	1.51	0.1661
MRATE=6				
DIFF	0.09976000	0.02267643	4.40	0.0017
MRATE=7				
DIFF	0.10064000	0.03233972	3.11	0.0125
MRATE=7.5				
DIFF	0.01072000	0.02588866	0.41	0.6885
MRATE=8				
DIFF	0.00374000	0.03304383	0.11	0.9124
MRATE=9				
DIFF	-0.03120000	0.01784514	-1.75	0.1143

------------------------ MRATE=10 ------------------------

DIFF	0.03172000	0.03806218	0.83	0.4262

------------------------ MRATE=11 ------------------------

DIFF	0.00050000	0.03126859	0.02	0.9876

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST D

VARIABLE	MEAN	STD ERROR OF MEAN	T	PR>¦T¦
MRATE=12				
DIFF	0.38077000	0.16277445	2.34	0.0441
MRATE=13				
DIFF	-0.03072000	0.06421847	-0.48	0.6438
MRATE=14				
DIFF	-0.06108000	0.06575663	-0.93	0.3772
MRATE=15				
DIFF	-0.03161000	0.08178488	-0.39	0.7081
MRATE=16				
DIFF	0.11210000	0.07773700	1.44	0.1832
MRATE=17				
DIFF	-0.12051000	0.06460055	-1.87	0.0950
MRATE=18				
DIFF	-0.16935000	0.10513094	-1.61	0.1417
MRATE=19				
DIFF	-0.42591000	0.15891465	-2.68	0.0252
MRATE=20				
DIFF	-0.50264000	0.15727101	-3.20	0.0109

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST D

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=0.5					
MOD3DEL	10	0.44064000	0.04059221	0.38800000	0.51800000
MOD2DEL	10	0.40802000	0.02163319	0.37810000	0.45620000
MRATE=1					
MOD3DEL	10	0.45792000	0.02410863	0.40900000	0.49310000
MOD2DEL	10	0.41887000	0.03127747	0.37320000	0.46930000
MRATE=2					
MOD3DEL	10	0.44599000	0.03486059	0.39630000	0.51120000
MOD2DEL	10	0.41962000	0.03262006	0.39010000	0.48880000
MRATE=3					
MOD3DEL	10	0.46813000	0.07427929	0.41290000	0.61600000
MOD2DEL	10	0.43538000	0.02561444	0.39250000	0.48390000
MRATE=4					
MOD3DEL	10	0.50404000	0.05667492	0.40740000	0.59610000
MOD2DEL	10	0.42786000	0.02355519	0.39500000	0.47280000
MRATE=5					
MOD3DEL	10	0.48621000	0.04478245	0.43090000	0.58510000
MOD2DEL	10	0.45206000	0.04631780	0.38930000	0.55140000
MRATE=6					
MOD3DEL	10	0.52305000	0.08084053	0.45110000	0.72840000
MOD2DEL	10	0.42329000	0.01867407	0.39870000	0.45390000
MRATE=7					
MOD3DEL	10	0.57686000	0.10911812	0.47290000	0.84200000
MOD2DEL	10	0.47622000	0.04888248	0.42190000	0.55010000
MRATE=7.5					
MOD3DEL	10	0.51136000	0.03744798	0.42730000	0.55440000

MOD2DEL	10	0.50064000	0.07314162	0.42040000	0.65670000

------------------------------ MRATE=8 ------------------------------

MOD3DEL	10	0.50362000	0.07178505	0.43280000	0.62730000
MOD2DEL	10	0.49988000	0.05470110	0.41310000	0.58210000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST D

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=9					
MOD3DEL	10	0.50844000	0.04930493	0.46150000	0.62690000
MOD2DEL	10	0.53964000	0.04378273	0.47690000	0.59930000
MRATE=10					
MOD3DEL	10	0.57365000	0.08470417	0.45340000	0.70900000
MOD2DEL	10	0.54193000	0.05246809	0.45520000	0.61470000
MRATE=11					
MOD3DEL	10	0.54014000	0.08919923	0.43080000	0.69650000
MOD2DEL	10	0.53964000	0.04378273	0.47690000	0.59930000
MRATE=12					
MOD3DEL	10	0.95286000	0.47060681	0.50200000	1.89200000
MOD2DEL	10	0.57209000	0.10842995	0.45640000	0.82820000
MRATE=13					
MOD3DEL	10	0.61065000	0.09390863	0.47750000	0.74100000
MOD2DEL	10	0.64137000	0.15198245	0.48430000	1.02200000
MRATE=14					
MOD3DEL	10	0.62963000	0.07497693	0.52060000	0.71770000
MOD2DEL	10	0.69071000	0.18819340	0.51950000	1.14410000
MRATE=15					
MOD3DEL	10	0.70162000	0.18803741	0.52040000	1.02400000
MOD2DEL	10	0.73323000	0.15322139	0.58330000	1.04100000
MRATE=16					
MOD3DEL	10	0.89471000	0.25745503	0.52720000	1.33800000
MOD2DEL	10	0.78261000	0.09159438	0.65060000	0.93250000
MRATE=17					
MOD3DEL	10	0.69352000	0.17374988	0.47110000	1.01400000

MOD2DEL	10	0.81403000	0.28564133	0.61050000	1.55100000

------------------------------ MRATE=18 ------------------------------

MOD3DEL	10	0.69765000	0.16744767	0.52280000	1.00300000
MOD2DEL	10	0.86700000	0.26178928	0.58570000	1.47700000

PAIRED-DIFFERENCE TEST
MESSAGE DELAY BY RATE
HOST D

VARIABLE	N	MEAN	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE
MRATE=19					
MOD3DEL	10	0.73652000	0.15388477	0.53910000	1.11300000
MOD2DEL	10	1.16243000	0.47167106	0.83410000	2.46200000
MRATE=20					
MOD3DEL	10	0.78041000	0.23682943	0.52620000	1.26300000
MOD2DEL	10	1.28305000	0.43460862	0.82170000	2.04900000

Bibliography

1. Benhamou, Eric and Judy Estrin. "Multilevel Internetworking Gateways: Architecture and Applications," IEEE Computer Magazine, 18: 27-34 (September 1983).

2. Bernstein, Mary M. Proposed DCEC IP Specification Internet Engineering Note: 186. Information Sciences Institute, University of Southern California, Marina del Rey, CA, June 1981.

3. Braden, R. and J. Postel. Requirements for Internet Gateways. Request for Comments: 1009. Information Sciences Institute, University of Southern California, Marina del Rey, CA, June 1987.

4. Cerf, Vinton G. and Edward Cain. "The DoD Internet Architecture Model," Computer Networks, 7: 307-318 (July'1983).

5. Cerf, Vinton G. and P. Kirstein. "Issues in Packet-Network Interconnection," Proceedings of the IEEE, 66: 1386-1408 (November 1978).

6. Gien, Michel and Hubert Zimmerman. "Design Principles for Network Interconnection," Sixth Data Communications Symposium. 109-119. New York: IEEE Press, 1979.

7. Hinden, Bob. A Host Monitoring Protocol. Request for Comments: 869. Information Sciences Institute, University of Southern California, Marina del Rey, CA, December 1983.

8. Hinden, Bob and Alan Sheltzer. The DARPA Internet Gateway. Request for Comments: 823. Information Sciences Institute, University of Southern California, Marina del Rey, CA, September 1982.

9. Kleinrock, Leonard. Queueing Systems, Volume II: Computer Applications. New York: John Wiley Sons, 1976.

10. Kleinrock, Leonard and William E. Taylor. "On Measured Behavior of the ARPA Network," Computer Networking, edited by Robert Blanc and Ira W. Cotton. IEEE Press, 1976.

11. Mills, Dave. Exterior Gateway Protocol Formal Specification. Request for Comments: 904. Information Sciences Institute, University of Southern California, Marina del Rey, CA, April 1984.

12. Nagle, John. On Packet Switches With Infinite Storage. Request for Comments: 970. Information Sciences Institute, University of Southern California, Marina del Rey, CA, December 1985.

13. Opderbeck, Holger and Leonard Kleinrock. "The Influence of Control Procedures on the Performance of Packet-Switched Networks," National Telecommunications Conference. 810-817. New York: IEEE Press, 1974.

14. Padlipsky, M. A. A Perspective on the ARPANET Reference Model. Request for Comments: 871. Information Sciences Institute, University of Southern California, Marina del Rey, CA, September 1982.

15. Pawlita, Peter F. "Traffic Measurement in Data Networks, Recent Measurement Results, and Some Implications," IEEE Transactions on Communications, 29: 525-535 (April 1981).

16. Postel, Jonathan, B. "Internetwork Protocol Approaches," Computer Network Architectures and Protocols, edited by Paul E. Green. New York: Plenum Press, 1982.

17. -----. Internet Protocol - DARPA Internet Program Specification. Request for Comments: 791. Information Sciences Institute, University of Southern California, Marina del Rey, CA, September 1981.

18. -----. Internet Control Message Protocol - DARPA Internet Program Protocol Specification. Request for Comments: 792. Information Sciences Institute, University of Southern California, Marina del Rey, CA, September 1981.

19. -----. Transmission Control Protocol - DARPA Internet Program Protocol Specification. Request for Comments: 793. Information Sciences Institute, University of Southern California, Marina del Rey, CA, September 1981.

20. -----. Simple Mail Transfer Protocol. Request for Comments: 821. Information Sciences Institute, University of Southern California, Marina del Rey, CA, August 1982.

21. ----- and Joyce Reynolds. Telnet Protocol Specification. Request for Comments: 854. Information Sciences Institute, University of Southern California, Marina del Rey, CA, May 1983.

22. ----- and Joyce Reynolds. File Transfer Protocol (FTP). Request for Comments: 959. Information Sciences Institute, University of Southern California, Marina del Rey, CA, October 1985.

23. ----- and Carl A. Sunshine. "The ARPA Internet Protocol," Computer Networks, 5: 261-271 (May 1981).

24. Pritsker, A. Alan B. Introduction to Simulation and SLAM II (Third Edition). New York: John Wiley Sons, 1986.

25. Prue, Walter and Jonathan Postel. Something a Host Could do with Source Quench: The Source Quench Introduced Delay (SQuID). Request for Comments: 1016. Information Sciences Institute, University of Southern California, Marina del Rey, CA, July 1987.

26. SAS Institute Inc. SAS User's Guide: Basics (Version 5 Edition). Cary, NC: SAS Institute Inc., 1985.

27. Sheltzer, Alan and others. "Connecting Different Types of Networks with Gateways," Data Communications, 12: 111-122 (August 1982).

28. Sollins, K. R. The TFTP Protocol (Revision 2). Request for Comments: 783. Information Sciences Institute, University of Southern California, Marina del Rey, CA, June 1983.

29. Stallings, William. Data and Computer Communications. New York: Macmillan Publishing Company, 1985.

30. Tanenbaum, Andrew S. Computer Networks. Englewood Cliffs: Prentice-Hall, Inc, 1981.

31. Zhang, Lixia. Congestion Control Fix for the ARPA Internet: A Congestion Control Algorithm for IP. Laboratory for Computer Science, Massachusetts Institute of Technology, draft paper.

VITA

Captain Bruce J. Schofield was born on 28 July 1949 in Gary, Indiana. He graduated from high school in Gary in 1967. In May 1971 he enlisted in the USAF as a Weather Equipment Technician. Six years later he was accepted into the Airman's Education and Commissioning Program (AECP). In May 1980, he received the degree of Bachelor of Science in Electrical Engineering from The University of Texas. He completed Officer Training School and was commissioned in August 1980. Upon completion of the Communications Engineer course at Keesler AFB, MS, he was assigned to the European Communications Division at Ramstein AB, GE. He served as a staff officer and finally as a Detachment Commander until entering the School of Engineering, Air Force Institute of Technology, in May 1896.

Permanent address: 4325 East 11 Place

Gary, IN 46403

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: Unclassified

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:

3. DISTRIBUTION / AVAILABILITY OF REPORT: [illegible]

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/ENG/87D-57

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Engineering

6b. OFFICE SYMBOL (If applicable): AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS (City, State, and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, OH 45433-6583

7b. ADDRESS (City, State, and ZIP Code):

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: Defense Communications Engineering Center

8b. OFFICE SYMBOL (If applicable): DCA/DCEC R640

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS (City, State, and ZIP Code): 1860 Wiehle Ave, Reston, VA 22090-5500

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification): EFFECTIVELY CONTROLLING DATAGRAM CONGESTION ON THE DOD INTERNET SYSTEM GATEWAYS

12. PERSONAL AUTHOR(S): Bruce J. Schofield, Captain, USAF

13a. TYPE OF REPORT: MS Thesis

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT (Year, Month, Day): 1987/Dec

15. PAGE COUNT:

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Internet, INternetworking, Congestion, Fair Queueing, Congestion Control, Gateway

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisor: LtCol Albert B. Garcia

Approved for public release: IAW AFR 190-1[illegible]
[signature] 1 Mar 88
[illegible] Development
[illegible] (ATC)
Wright-Patterson AFB OH [illegible]

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL: Albert B. Garcia

22b. TELEPHONE (Include Area Code): (513) 255-

22c. OFFICE SYMBOL: AFIT/ENG

Under the current implementation of the DoD Internet, a gateway's response to congestion is to discard datagrams. Discarding datagrams increases message delay and wastes network resources. Several congestion control methods have been proposed to improve the performance of the Internet. This study looked at two; Nagle's Fair queueing and Zhang's Metered queueing.

Nagle proposes to replace the single queue per outgoing channel with multiple queues, one for each source with datagrams passing through the gateway. Datagrams are removed from these queues one at a time in a round robin fashion. This procedure ensures each source is allotted a fair share of the channel bandwidth. The study found, through simulation, that this method insulated well behaved host from the presence of a badly behaved host. Badly behaved host are in effect punished through increased delay while well behaved host receive their fair share of the network resources. This researcher recommends Nagle's method be implemented for testing on the Internet.

Zhang proposal is basically a feedback method of congestion control. This method allows a gateway to control the rate at which host send datagrams through the gateway. This requires modification to the IP modules in the hosts and gateways and modification to the Source Quench message. These modifications will allow the gateways to sense traffic levels and to tell the host what rate to transmit at and for how long. However, Zhang did not define two parameters which are critical to the performance of her method. Both of these parameters depend on the Internet traffic profile which is not known at the present. Because these parameters are not defined, this study could not simulate the performance of Zhang's method.

END
DATE
FILMED
4-88
DTIC